目　录

第1章 绪　　论

本章介绍了 MATLAB 的基本特性及其基本语法。

1.1 概述

MATLAB 的名称源自 *Matrix Laboratory*，是一门计算语言，它专门以矩阵的形式处理数据。MATLAB 将计算与可视化集成到一个灵活的计算机环境中，并提供了大量的内置函数，可以在广泛的工程问题中直接利用这些函数获得数值解。

1.2 MATLAB 的使用方法

MATLAB 启动后，显示一个窗口并在“>>”提示符后显示闪烁[1] 的光标。这个窗口通常称为 MATLAB 命令窗口。

命令窗口作为一个工作空间等同于一张空白的工作簿。“>>”提示符之后的空间称为命令行。当输入数值矩阵时，只须在命令行中按第 2.4 节定义的格式输入矩阵数值即可。要想将这些数值赋给变量，可在数值前加变量名及等号。如果未给出变量名，则 MATLAB 将数值赋给通用的变量名 *ans*。这样，在 MATLAB 命令窗口调用和显示这些数据时或将其应用于 MATLAB 表达式中时，输入用户自定义的变量名或默认变量名 *ans* 即可。但是，如果用相同的变量名定义了不同的数值矩阵或在未声明变量名的前提下输入了另一组数值，则原先输入的数值将被覆盖掉。要想清除工作空间，可输入 clear 命令。关于 clear 命令将在下面详细讨论。

使用 MATLAB 如同用计算器一样，它可以对变量进行算术运算、三角函数运算和幂运算（例如加法、除法、余弦和对数运算等）。这些运算可由 MATLAB 调用不同的函数完成。除基本的运算函数外，MATLAB 还提供了大量的函数来完成复杂的运算。同时，MATLAB 还支持用户创建函数，详见第 5 章。这些函数的另一个用途是结构化编程。它们不同于命令行下输入的表达式，在此种情况下，MATLAB 为它们分配专有的操作空间并且预定义一些与 MATLAB 环境相关的输入输出关系。

在命令窗口中输入多个表达式或重复输入一组表达式将会使用户感到很麻烦。为避免这一问题，MATLAB 定义了程序文件——包含一组命令的文件，文件中每一行命令的执行都同它们在命令窗口里执行一样。程序文件可由文字处理器、文本编辑器或由 MATLAB 提供的文本编辑器及调试器[2] 创建，以文本文件形式保存，扩展名为“.m”。如果程序文件是由文字处理器或文本编辑器创建的，则在 MATLAB 命令窗口中仅输入不带扩展名的文件名即可执行。如

---

① 关于 MATLAB 窗口风格、窗口管理及文件管理的说明与 Windows 环境有关。其他的操作系统具有相似的处理过程。

② 单击 MATLAB 命令窗口最左边的图标（即白色矩形图标）或从 File 下拉菜单中选择 New 或 M-File 子菜单，即可打开 MATLAB 编辑/调试窗口。

果程序文件是由 MATLAB 编辑器创建的，则可单击 Tools 菜单，选择 Run 子菜单执行程序文件。这要求文件在执行前必须已被保存。

在如下条件下可采用程序文件的形式：

1. 程序包含多行代码
2. 程序将重复使用
3. 希望永久保存
4. 要求随时更新
5. 需要进行大量调试
6. 需要将程序清单分发给其他人或组织

此外，程序或函数通常具备如下属性：

1. **说明部分**，最少应包含如下内容：

   程序用途及其执行的操作
   程序员名
   创建日期
   修改日期
   输入量描述：数值、含义及类型
   输出量描述：数值、含义及类型

2. **输入部分**，对输入值进行全面检验，保证所有输入值的正确性，使程序或函数能正确执行。
3. **初始化部分**，为变量分配初始数值。
4. **计算部分**，完成主要的数值计算。
5. **输出部分**，计算结果以图形方式或格式化方式输出。

1.3 创建变量名

MATLAB 允许用户创建的变量名不可超过 31 个字符，多余部分将被忽略掉。变量名要求以大写或小写字母开头，后面可跟大小写字母、数字或下划线。字符间不允许有空格。变量名对于大小写敏感，即变量名 *junk* 与 *junK* 是不同的。MATLAB 中存在两个一般约定：使用下划线和首字母大写。例如，exit pressure 是所要计算的数值，那么在 MATLAB 命令行、程序或函数中它可被定义为 *exit_pressure* 或 *ExitPressure*。

本章结尾处附录 A 的表 A.1 给出了 MATLAB 中最常使用的字符、符号及其含义。

1.4 变量管理

在 MATLAB 运行期间(即程序退出之前)，除非调用 clear 函数，否则 MATLAB 会在内存中保存全部变量值，包括命令行输入的变量以及执行程序文件所引入的变量。clear 函数可清除内存中的所有变量。如前所述，在 MATLAB 运行期间可随时通过输入变量名或从表达式中引用变量名来访问变量的当前数值(在未调用 clear 函数之前)。这些变量可作为全局变量加以引用。

MATLAB命令窗口输入的信息会保持在窗口中,并可通过滚动条重新访问。一旦信息量超出其滚动内存容量,则最早输入的信息将会丢失。虽然程序文件中所引入的表达式不能被重新访问,但程序文件中的变量名和变量值却可重新访问。输入表达式的记录可通过单击MATLAB命令窗口顶部Edit下拉菜单中的Clear Session子菜单加以清除。但应注意,该操作仅清除MATLAB的命令窗口内容而不能删除变量。删除变量只能通过使用clear来实现。此外,copy和paste图标可用来实现将MATLAB命令窗口中以前输入的表达式复制到当前行或将MATLAB表达式由MATLAB命令窗口粘贴到文字处理器窗口,反之亦然。

在MATLAB命令窗口中输入whos命令或选择File菜单的Show Workspace子菜单,可以创建一个变量信息显示窗口,在此窗口中显示变量列表。这些变量是自上次调用clear函数后产生的。显示内容为变量名、变量长度、每个变量所占用的存储字节数及变量类型。变量类型包括:数值型、字符串型(详见第3.1节)、符号型及inline对象(详见第5.3节)等。后一种方法的优点在于如果使变量信息窗口保持在激活状态,则MATLAB会对其进行自动更新。

1.5 访问程序与函数文件

程序文件及函数文件可通过在MATLAB命令窗口中输入其文件名(不带".m"扩展名)加以执行。但MATLAB首先要知道文件保存的路径。路径信息的输入可通过File下拉菜单的Set Path子菜单实现。单击Set Path子菜单,打开Path Browser窗口,再单击Browse按钮选择目标目录,然后单击Path \ Add to Path菜单并选定Add to Front或Add to Back选项以添加目标路径。在Path Browser窗口可添加多个路径。退出Path Browser窗口前请选择File菜单的Save Path子菜单,该操作将会保存添加的路径以备下次应用。

1.6 命令窗口管理

为在命令窗口中更加清晰地显示出字母及数字,MATLAB提供了format函数的几种功能选项。其中最为常用的是:

```
format compact
```

和

```
format long e
```

前者剔除了显示中多余的空行或空格,后者可将系统默认的5位数字的显示格式转换为16位数字位加3位指数位的显示格式。当调试程序时,产生的数值变化量很小或者数值变化范围很大的情况下,采用format long e选项尤为重要。要恢复系统默认设置,可输入:

```
format short
```

如想更改MATLAB命令窗口显示的字体及大小,单击File下拉菜单的Preferences子菜单,在弹出的参数设置窗口中选择Command Window Font选项卡并设定各选择项。

MATLAB中十分有用的两个组合键为"^c"(即Ctrl键与c键同时按下)和"^p"(即Ctrl键与p键同时按下)。"^p"用于将最近一次键入的信息显示在MATLAB命令窗口中,按Enter键可再

次执行该命令。按 Enter 键执行前,可修改调入的命令行。如果连续两次按下“^p”组合键,可调用在上次键入的命令行,依此类推。

完成同样的功能也可通过上、下箭头键实现。

“^c”用于终止程序或函数的执行,也可用于退出暂停的程序或函数。

1.7 向命令窗口输入程序和函数

在 MATLAB 命令窗口中输入数值(若为向量或矩阵,则应输入一组数值,详见第 2.3 节和第 2.4 节)时,程序或函数中应包含如下语句:

```
VariableName = input('Any message')
```

其中,input 是 MATLAB 的函数,而 *Any message* 是显示在 MATLAB 命令窗口中的信息。指令执行后,输入相应数据并按下 Enter 键后,输入的数值(或一组数值)被赋给变量 *VariableName*。其他输入数据的方法可参见第 3.3 节。另外,第 3.2 节更为详尽地描述了 input 函数的使用方法。

此外,还有一种方法可使程序结果输出到 MATLAB 命令窗口中。首先,可在表达式后面省略分号,见第 1.8 节(还可参见本章附录 A 的表 A.1)。采用这种方法,MATLAB 可在命令窗口中显示变量名,后跟等号,并在下一行显示变量数值。这种方法在调试过程中极为有用。此外,如想使输出数值的格式显示得更加统一,可采用:

```
disp
```

或

```
fprintf
```

第 3.1 节对此进行了讨论。

1.7.1 在线帮助

MATLAB 拥有完整的在线帮助,可通过几种方法对它进行访问。一种是单击工具栏上的问号图标,打开 Help 窗口。为便于从 MATLAB 窗口或文字处理器/文本编辑器窗口引用在线帮助,每次查询后,应将 Help 窗口最小化。另一种使用在线帮助的方法是在 MATLAB 窗口中键入:

```
help FunctionName
```

其中,*FunctionName* 为要查询的函数名。

第三种方法是从 Help 下拉菜单中选择 Help Desk (HTML)子菜单,它将通过操作系统的 Web 浏览器显示完整的帮助命令窗口。此外,还可以通过 MATLAB 命令窗口中的“?”图标打开 Help 窗口,再单击 Go to Help Desk 按钮来访问 Web 浏览器,并显示完整的帮助命令窗口。

1.8 MATLAB 的基本语法

MATLAB 要求表达式在使用变量以前一定要先对其进行赋值操作(符号工具箱使用的变

量除外)。对变量赋值时,先输入变量名和等号,然后输入变量数值并按 Enter 键结束。如想实现 $p=7.1, x=4.92, k=-1.7$ 功能,必须要在 MATLAB 命令窗口中进行如下的信息交互:

```
>>p=7.1          ←—用户输入
p=
  7.1000         ←—系统响应
>> x=4.92        ←—用户输入
x=
  4.9200         ←—系统响应
>> k=-1.7        ←—用户输入
k=
 -1.7000         ←—系统响应
```

在表达式后加分号(;)可省略系统的响应信息,如:

```
>> p=7.1;
>> x=4.92;
>> k=-1.7;
>>
```

MATLAB 允许在一行中输入多个表达式,行尾以按下 Enter 键结束。表达式间以逗号或分号进行分隔。使用逗号分隔表达式时系统会回显输入的数值。例如按如下格式输入信息:

```
>> p=7.1 , x=4.92 , k=-1.7
```

系统显示为:

```
p=
  7.1000
x=
  4.9200
k=
  -1.7000
>>
```

如用分号分隔表达式,则系统不会输出响应信息。

标量的加、减、乘、除和幂运算由 5 个运算符号表示,它们分别是 + 、- 、* 、/ 和^ 。例如,算术表达式:

$$t=\left(\frac{1}{1+px}\right)^k$$

在 MATLAB 中可写为:

```
t=(1/(1+p*x))^k
```

在执行本语句之前首先要对 p、x 及 k 赋值,否则会显示错误信息。p, x, k 正确赋值后执行本语句,系统会显示计算结果:①

```
440.8779
```

① 在以后的章节中,不再重复声明"在命令窗口中",而只是给出对应格式的输出结果(或应答)。

在变量 t 的 MATLAB 表达式中使用了圆括号,用来确定运算次序。MATLAB 本身包含默认的运算次序,故可最大限度地减少圆括号的使用数目。为了使表达式更易理解可增加一些额外的圆括号。默认的运算次序为:幂运算为最高,其次为乘除,最后为加减。不管是在成对的圆括号内,还是在通常的表达式中,MATLAB 遵循从左到右的计算次序。表 1.1 是一个实例,对数学表达形式与 MATLAB 表达形式之间的差异进行比较,其中包含了标量数值 c, d, g, x。

表 1.1 MATLAB 语法举例

数学表达式	MATLAB 表达式
$1-dc^{x+2}$	1-d*c^(x+2)
dc^x+2	d*c^x+2 或 2+d*c^x
$(2/d)c^{x+2}$	(2/d)*c^(x+2) 或 2/d*c^(x+2) 或 2*c^(x+2)/d
$(dc^x+2)/g^{2.7}$	(d*c^x+2)/g^2.7
$\sqrt{dc^x+2}$	(d*c^x+2)^0.5 或 sqrt(d*c^x+2)

MATLAB 函数 sqrt 用于求解输入变量的平方根。

MATLAB 也包含大量的基本及非基本函数。表 1.2 和表 1.3 列出了一些基本函数,这些函数可用于标量、向量或矩阵的运算。MATLAB 中向量和矩阵的定义及创建,参见第 2.3 节。

表 1.2 基本函数

数学表达式	MATLAB 表达式
e^x	exp(x)
$\sqrt{x}$	sqrt(x)
$\ln(x)$ 或 $\log_e(x)$	log(x)
$\log_{10}(x)$	log10(x)
$\|x\|$	abs(x)
符号函数	sign(x)

表 1.3 MATLAB 的三角函数和双曲函数

	三角函数		双曲函数	
函数	函数	反函数	函数	反函数
正弦	sin(x)	asin(x)	sinh(x)	asinh(x)
余弦	cos(x)	acos(x)	cosh(x)	acosh(x)
正切	tan(x)	atan(x)[①]	tanh(x)	atanh(x)
正割	sec(x)	asec(x)	sech(x)	asech(x)
余割	csc(x)	acsc(x)	csch(x)	acsch(x)
余切	cot(x)	acot(x)	coth(x)	acoth(x)

① atan2(y,x)为四象限形式。

下面的表达式举例说明了 MATLAB 内置函数的使用方法:

$$y=\sqrt{|\pi-\sin(x)/\cosh(a)-\ln_e(x+a)|}$$

在 MATLAB 中该表达式可写为:

```
y = sqrt(abs(pi - sin(x)/cosh(a) - log(x + a)))
```

其中,内置函数 pi = π,并且在本语句前已对变量 x 和 a 赋了值。

1.9 使用 MATLAB 的一些建议

下面给出几点关于如何在 MATLAB 环境中高效创建程序和函数的建议。

- **尽可能使用帮助文件。**它将减少语法错误,以及由于不正确或不恰当地引用 MATLAB 函数而产生的错误。
- **在文本编辑器中编写程序和函数并保存为".m"文件。**这将节省时间,减少代码。当使用 MATLAB 编辑器/调试器时,将极大地简化调试过程。
- **尽量减少由程序或函数组成的表达式的数目。**在易读性和简洁性之间采用折中的办法,并充分考虑到函数和调用过程的执行速度和直接性。
- **程序和函数应尽可能使用图形化开发手段。**这种方法可检测出隐含的代码错误,从而缩短代码的开发过程,并可使被分析和模型化的物理过程变得容易理解。
- **最为重要的是要采用另一种手段验证程序或函数输出结果的正确性。**

练习

1.1 下列表达式[①] 表达了在主接触应力 F 的作用下,两个球体被挤压到一起时在 x, y, z 方向上所产生的接触应力值。

$$\sigma_x = \sigma_y = -p_{max}\left[\left(1 - \frac{z}{a}\arctan\left(\frac{a}{z}\right)\right)(1 - v_1) - 0.5\left(1 + \frac{z^2}{a^2}\right)^{-1}\right]$$

$$\sigma_z = \frac{-p_{max}}{1 + z^2/a^2}$$

式中,

$$a = \sqrt[3]{\frac{3F}{8}\frac{(1 - v_1^2)/E_1 + (1 - v_2^2)/E_2}{1/d_1 + 1/d_2}}$$

$$p_{max} = \frac{3F}{2\pi a^2}$$

v_j、E_j 和 $d_j(j = 1,2)$分别为两个球体的泊松比、弹性模量和直径。

用 MATLAB 表达形式写出上述公式并用下列给出的数值进行计算:$v_1 = v_2 = 0.3$, $E_1 = E_2 = 3 \times 10^7$, $d_1 = 1.5$, $d_2 = 2.75$, $F = 100$ lb(磅,等于 373 克), $z = 0.01$ in(英寸,等于 2.54 厘米)。保存这些表达式以备在练习 6.4 中绘制图形。

[答案: $a = 0.013\,0$ in, $p_{max} = 281\,580$ psi(磅/平方英寸), $\sigma_x = -108\,580$ psi, $\sigma_z = -177\,120$ psi。]

---

① 参见 J. E. Shigley 和 C. R. Mischke, *Mechanical Engineering Design*, 第五版, McGraw-Hill, New York, 1989 年。

1.2 下列表达式[①] 表示了在主接触应力 F 的作用下,两个平行圆柱体被挤压到一起时在 x,y,z 方向所产生的接触应力值。

$$\sigma_x = -2v_2 p_{max}\left(\sqrt{1+\frac{z^2}{b^2}}-\frac{z}{b}\right)$$

$$\sigma_y = -p_{max}\left(\left(2-\left(1+\frac{z^2}{b^2}\right)^{-1}\right)\sqrt{1+\frac{z^2}{b^2}}-2\frac{z}{b}\right)$$

$$\sigma_z = \frac{-p_{max}}{\sqrt{1+z^2/b^2}}$$

$$\tau_{yz} = 0.5(\sigma_y - \sigma_z)$$

式中,

$$p_{max} = \frac{2F}{\pi bL}$$

$$b = \sqrt{\frac{2F}{\pi L}\frac{(1-v_1^2)/E_1+(1-v_1^2)/E_2}{1/d_1+1/d_2}}$$

v_j、E_j 和 $d_j(j=1,2)$分别为两个球体的泊松比、弹性模量和直径。

用 MATLAB 表达形式写出上述公式并用下列给出的数值进行计算:$v_1=v_2=0.3$,$E_1=E_2=3\times10^7$,$d_1=1.5$,$d_2=2.75$,$F=100$ lb,$L=2$,$z=0.001$ in。保存这些表达式以备在练习 6.5 中绘制图形。

[答案:$b=0.001\ 4$ in,$p_{max}=23\ 251$ psi,$\sigma_x=-7\ 085.7$ psi,$\sigma_y=-4\ 843.8$ psi,$\sigma_z=-18\ 775$ psi。]

1.3 流体轴承的载荷数由下式给出:[②]

$$N_L = \frac{\pi\varepsilon\sqrt{\pi^2(1-\varepsilon^2)+16\varepsilon^2}}{(1-\varepsilon^2)^2}$$

式中 ε 为偏心率。

用 MATLAB 表达形式写出上式并用给出的 $\varepsilon=0.8$ 进行计算。

[答案:$N_L=72.022$。]

1.4 高为 h 的螺栓,材料的弹性模量为 E。当它通过直径为 d_0 的螺孔时螺栓刚度可由下式得出:[③]

$$k = \frac{\pi E d_0 \tan 30°}{\ln\dfrac{(d_2-d_0)(d_1+d_0)}{(d_2+d_0)(d_1-d_0)}}$$

式中 d_1 为孔下垫圈的直径,且:

$$d_2 = d_1 + h\tan 30°$$

---

① 参见 J. E. Shigley 和 C. R. Mischke,出处同上。

② R. L. Norton, *Machine Design, An Integrated Approach*, Prentice-Hall, Upper Saddle River, NJ, 1996 年。

③ A. H. Burr 和 J. B. Cheatham, *Mechanical Analysis and Design*, 第二版, Prentice-Hall, Upper Saddle River, NJ, 1995 年,第 423 页。

用 MATLAB 表达形式写出该方程。注意在 MATLAB 中三角函数的参数单位应为弧度。用下列值进行计算：$h=1.25$，$d_0=0.25$，$d_1=0.625$，$E=3\times10^7$。

[答案：$d_2=1.346\,7$ in，$k=2.884\,2\times10^7$ lb/in。]

1.5 一个长管，其内表面半径为 a，温度为 T_a；外表面半径为 b，温度为 T_b，则其径向和切向应力可分别表示为：①

$$\sigma_r=\frac{\alpha E(T_a-T_b)}{2(1-v)\ln(b/a)}\left[\frac{a^2}{b^2-a^2}\left(\frac{b^2}{r^2}-1\right)\ln\left(\frac{b}{a}\right)-\ln\left(\frac{b}{r}\right)\right]$$

$$\sigma_t=\frac{\alpha E(T_a-T_b)}{2(1-v)\ln(b/a)}\left[1-\frac{a^2}{b^2-a^2}\left(\frac{b^2}{r^2}+1\right)\ln\left(\frac{b}{a}\right)-\ln\left(\frac{b}{r}\right)\right]$$

式中 r 为管子的径向坐标，E 为管子材料的弹性模量，α 为热膨胀系数。

沿径向管壁的温度分布为：

$$T=T_b+\frac{(T_a-T_b)\ln(b/r)}{\ln(b/a)}$$

用 MATLAB 表达形式写出该方程并用如下值进行验证：$\alpha=1.2\times10^{-5}$，$E=3\times10^7$，$v=0.3$，$T_a=500$，$T_b=300$，$a=0.25$，$b=0.5$，$r=0.375$。

[答案：$\sigma_r=-8\,011.5$，$\sigma_t=5\,231.9$，$T=383.007\,5$。]

1.6 本金 P 以每年 n 次，每次 $i\%$ 的增值率（n 与 i 的乘积为每年增值额的百分比）增加，当增加到 r_pP 时所花费的时间为：

$$T=\frac{\ln r_p}{n\ln(1+0.01i)}\ \text{年}$$

用 MATLAB 表达形式写出该方程并用如下给出值验证：$r_p=2$，$i=0.5$，$n=12$。

[答案：$T=11.581\,3$ 年。]

1.7 压力为 p_0 的气体在可逆绝热条件下从容器中溢出，其质量与流量之比如下：②

$$\Psi=\sqrt{\frac{k}{k-1}}\sqrt{\left(\frac{p_e}{p_0}\right)^{2/k}-\left(\frac{p_e}{p_0}\right)^{(k+1)/k}}$$

式中 p_e 为容器出口的外部压力，k 为可逆绝热的气体常数。

用 MATLAB 表达形式写出该方程并用如下给出值进行验证：$k=1.4$，$p_e/p_0=0.3$。

[答案：$\Psi=0.427\,1$。]

1.8 抛物线型横截面开口管道流动的流量系数为：③

$$K=\frac{1.2}{x}\left[\sqrt{16x^2+1}+\frac{1}{4x}\ln\left(\sqrt{16x^2+1}+4x\right)\right]^{-2/3}$$

式中 x 为最大水深与液体表面处管道宽度的比值。

用 MATLAB 表达形式写出该方程并用如下给出值进行验证：$x=0.45$。

[答案：$K=1.339\,4$。]

---

① A. H. Burr 和 J. B. Cheatham，出处同上，第 496 页。

② 参见 W. Beitz 和 K. H. Kuttner 编辑，*Handbook of Mechanical Engineering*，Springer-Verlag，New York，1994 年，第 C15 页。

③ 参见 H. W. King，*Handbook of Hydraulics*，第四版，McGraw-Hill，NY，1954 年，第 7~24 页。

1.9 依据如下给出的公式①，当 $n=0$ 时可在 10^{-7} 的误差范围内算出 π 的近似值，而当 n 取两项（$n=0,1$）时其误差范围则为 10^{-15}。实际上，n 的取值每增加一个，π 近似值的精度将提高 10^{-8}。因此，n 取 4 项（$n=0,1,2,3$）时 π 值可计算到 31 位，可用 MATLAB 符号工具箱进行验证。

$$\frac{1}{\pi}=\frac{\sqrt{8}}{9\,801}\sum_{n=0}^{\infty}\frac{(4n)!(1\,103+26\,390n)}{(n!)^4 396^{4n}}$$

注意，阶乘的计算可采用下述 gamma 函数：gamma$(n+1)=n!$；gamma$(4n+1)=(4n)!$，$n=0,1,\cdots$。

附录 A MATLAB 特殊字符汇总

表 A.1 MATLAB 特殊字符及使用说明

符号	名称	用途
.	句点	(a)小数点 (b)向量或矩阵的一种操作类型，称为点操作，例如 $c=a.*b$.
,	逗号	(a)参数分隔符，矩阵元素分隔，如 $b(2,7)$；函数参数分隔，如 besselj(1,x)；创建向量参数分隔，如 $v=[1,x]$；函数参数输出分隔，如[x,s] = max(a) (b)当几个表达式在同一行时放在每个表达式之后
;	分号	(a)放在一个表达式末尾处或当几个表达式在同一行时放在每个表达式末尾处，不显示计算结果 (b)在创建矩阵的语句中指示一行的结束，例如 $m=[x\ y\ z;a\ b\ c]$
:	冒号	(a)创建向量的表达式分隔符，如 x = a:b:c (b)对矩阵 z 而言，当写为 $z(:,k)$ 时表示所有行；当写为 $z(k,:)$ 时表示所有列
()	圆括号	(a)矩阵 z 的某一个元素的下标指示，如 $z(j,k)$ 表示矩阵的 j 行 k 列的元素 (b)算术表达式分隔符，如 $a^{\wedge}(b+c)$ (c)函数参数分隔符，如 sin(x)
[]	方括号	创建一组数值、向量、矩阵或字符串（字母型）
{}	大括号	创建单元矩阵或结构
%	百分号	注释分隔符，用于指示注释的开始，MATLAB 编译器会忽略其右边的内容，但当用于在一对引号内部定义字符串时除外，如 a = 'pl = 14 % of the total'
'	引号或撇号	(a)'Expression'表明 Expression 为字符串（字母型） (b)表示向量或矩阵的转置
...	省略号	表示 MATLAB 表达式继续到下一行，增强代码可读性
	空格	上下文相关：或者忽略从而作为数据创建语句的分隔符，如 c = [a b]，或者作为字符串语句的一个字符

① S. Ramanujan, "Modular equations and approximations to π," *Quart. J. Math.*, 第 45 卷，第 350 ~ 372 页。

第 2 章　矩阵与 MATLAB

本章将通过向量和矩阵及它们的运算来说明 MATLAB 的基本语法。

2.1　概述

MATLAB 是一门语言，它的运算指令和语法基于一系列基本的矩阵运算以及它们的扩展运算。因此，为了更好地利用 MATLAB 语言的优越性和简洁性，首先对基本矩阵的定义、符号体系做一概述，然后给出几个应用实例。本部分所论述的内容为基础内容，在本章及后续章节中将会大量应用。

2.2　矩阵和向量

由 m 行 n 列构成的数组 $\boldsymbol{a}$ 称为($m\times n$)阶矩阵，它总共由($m\times n$)个元素组成并按如下的形式排列：

$$\boldsymbol{a}=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & \\ \vdots & & \ddots & \\ a_{m1} & & & a_{mn} \end{bmatrix}\rightarrow(m\times n)$$

矩阵的元素记为 a_{ij}，其中 i 表示行，j 表示列。矩阵 $\boldsymbol{a}$ 的大小可用如下命令获得：

```
size(a)
```

或

```
[m,n] = size(a)
```

第一种形式中，size 可显示出两个值：第一个值为行数(m)，第二个值为列数(n)。对于第二种用法，[m, n]表示矩阵的行数赋给 m，列数赋给 n。更详细的介绍见第 5.2 节。

矩阵 $\boldsymbol{a}$ 的几种特例论述如下。

2.2.1　方阵

当 $m=n$ 时，$\boldsymbol{a}$ 称为方阵。

2.2.2　对角阵

当 $a_{ij}=0, i\neq j$，且 $m=n$ 时，得到对角阵：

$$a=\begin{bmatrix}a_{11} & 0 & \cdots & 0\\ 0 & a_{22} & & \\ \vdots & & \ddots & \\ 0 & & & a_{nn}\end{bmatrix}\rightarrow(n\times n)$$

因此,如果 $a=[1\ 3\ 8]$(见第 2.3 节向量的产生),那么,

```
a=[1 3 8];
z=diag(a)
```

产生如下对角阵:

```
z =
    1  0  0
    0  3  0
    0  0  8
```

这是 MATLAB 中显示的(3×3)阶矩阵:

$$\begin{bmatrix}1 & 0 & 0\\ 0 & 3 & 0\\ 0 & 0 & 8\end{bmatrix}$$

另外,如果 b 是一个方阵,那么,

```
diag(b)
```

返回 b 的对角元素。

当 $a_{ii}=1$ 并且 $m=n$ 时,产生单位矩阵 I,记做:

$$I=\begin{bmatrix}1 & 0 & \cdots & 0\\ 0 & 1 & & \\ \vdots & & \ddots & \\ 0 & & & 1\end{bmatrix}$$

单位矩阵的 MATLAB 表达式为:

```
eye(n)
```

其中 n 为方阵的大小。因此,MATLAB 表达式:

```
c=eye(3)
```

产生:

```
c =
    1  0  0
    0  1  0
    0  0  1
```

2.2.3 列矩阵和行矩阵(向量)

当 $a_{ij}=a_{i1}$(即只有 1 列时),a 称为列矩阵或者列向量,记做:

$$a=\begin{bmatrix}a_{11}\\a_{21}\\\vdots\\a_{m1}\end{bmatrix}=\begin{bmatrix}a_1\\a_2\\\vdots\\a_m\end{bmatrix}\to(m\times 1)$$

然而,当 $a_{ij}=a_{1j}$ 时,即矩阵只有 1 行,则矩阵称为行矩阵或者行向量,记做:

$$a=[a_{11}\quad a_{12}\quad\cdots\quad a_{1n}]=[a_1\quad a_2\quad\cdots\quad a_n]\to(1\times n)$$

在 MATLAB 中,这是向量的默认定义。

2.2.4　矩阵和向量的转置

矩阵的转置用(′)表示,定义如下:

$$a=\begin{bmatrix}a_{11}&a_{12}&\cdots&a_{1n}\\a_{21}&a_{22}&&\\\vdots&&\ddots&\\a_{m1}&&&a_{mn}\end{bmatrix}\to(m\times n)$$

如果 a 是一个$(m\times n)$阶矩阵,那么它的转置矩阵 $w=a'$ 为如下的$(n\times m)$阶矩阵:

$$w=a'=\begin{bmatrix}w_{11}=a_{11}&w_{12}=a_{21}&\cdots&w_{1m}=a_{m1}\\w_{21}=a_{12}&w_{22}=a_{22}&&\\\vdots&&\ddots&\\w_{n1}=a_{1n}&&&w_{nm}=a_{mn}\end{bmatrix}\to(n\times m)$$

对列向量和行向量,如果:

$$a=\begin{bmatrix}a_1\\a_2\\\vdots\\a_m\end{bmatrix}\to(m\times 1)\quad 那么,a'=[a_1\,a_2\cdots a_m]\to(1\times m)$$

如果:

$$a=[a_1\,a_2\cdots a_m]\to(1\times m)\quad 则,a'=\begin{bmatrix}a_1\\a_2\\\vdots\\a_m\end{bmatrix}\to(m\times 1)$$

向量 a 的长度,即元素的个数,可以通过下面的命令获得:

```
L= length(a)
```

或

```
L= size(a)
```

然而矩阵的大小只能通过命令 size 获得。使用 size 命令的好处是不需要事先知道 a 是向量还是矩阵。但是对于向量来说,length 命令使用更方便。

2.3 向量的创建

向量可以表示为：

$$f=[a\ x\ b\ \cdots]\ \text{或}\ f=[a,x,b,\ \cdots]$$

其中 $a,x,b,\cdots$ 可以是变量、数值、表达式或字符串(见第 3.2 节)。如果它们是变量或表达式，那么所有变量及由这些变量所构成的表达式必须预先定义。并且在执行语句之前，每一个变量必须先赋值。表达式和变量可以任何组合和任何次序出现。对于采用下列形式的情况：

$$f=[a\ x\ b\ \cdots]$$

符号之间必须要用空格隔开，然而对于采用下列形式的情况：

$$f=[a,x,b,\ \cdots]$$

符号间是由逗号隔开的，空格可以省略。

应着重说明的是，如果 a 是一个表达式，则表达式字符和运算符之间可无空格。例如，如果 $a=h+d^g$，那么 f 可以写为：

$$f=[h+d\hat{}g\ x\ b\ \cdots]$$

或

$$f=[h+d\hat{}g,x,b,\ \cdots]$$

在 MATLAB 中给出了几种方法，用来对向量和矩阵中的元素进行赋值。矩阵的创建方法见第 2.4 节。第一种方法是使用冒号来指定数值范围和相邻值的步长，第二种方法是指定数值的范围和期望的数值个数。在前一种方法中，步长最为重要。而后一种方法中数值的个数最为重要。利用冒号创建向量的方法如下：

```
x = s:d:f
```

其中，

s = 起始值或初始化值

d = 增量或减量值

f = 结束值或终值

因此，可产生如下的行向量 $\boldsymbol{x}$：

$$\boldsymbol{x}=[s\ s+d\ s+2d\ \cdots\ s+nd]$$

其中 $s+nd\leqslant f$。应注意到在创建向量 $\boldsymbol{x}$ 时没有直接指定 n 的数值。s、d 和 f 可以是数值、变量和表达式的任意组合。向量 $\boldsymbol{x}$ 中元素的个数由下式确定：

```
n = length(x)
```

当 d 省略时，MATLAB 默认 $d=1$。即：

```
x = s:f
```

产生向量：

$$\boldsymbol{x}=[s,s+1,s+2,\ \cdots\ ,\ s+n]$$

其中 $s+n\leqslant f$。同样，s 和 f 可以是数值、变量和表达式的任意组合。

另一方面，可以指定 n 为从 s 到 f 的等间隔值，如下所示：

```
x = linspace(s,f,n)
```

其中增量或减量值 d 由 MATLAB 通过下式计算得出：

$$d = \frac{f - s}{n - 1}$$

s 和 f 的值可以为正也可以为负，并允许 $s > f$ 或 $s < f$。如果没有指定 n 的值，则 MATLAB 会给出一个默认值 $n = 100$。因此，linspace 可创建如下的向量：

$$\boldsymbol{x} = [s \ s+d \ s+2d \ \cdots \ f = s+(n-1)d]$$

如果需要用对数等间隔，则：

```
x = logspace(s,f,n)
```

其中 10^s 为初值，10^f 为终值，d 的定义同上。因此可创建如下向量：

$$\boldsymbol{x} = [10^s \ 10^{s+d} \ 10^{s+2d} \cdots \ 10^f]$$

在此，当 n 没有给出时，MATLAB 给出默认值 $n = 50$。

为进一步说明 MATLAB 中的向量，令：

$$\boldsymbol{b} = [b_1 \ b_2 \ b_3 \cdots \ b_n]$$

即创建一个一行 n 列的向量 $\boldsymbol{b}$。在 MATLAB 中，利用写入 $\boldsymbol{b}(3)$ 的方式获得向量 $\boldsymbol{b}$ 中 b_3 的值。即 $\boldsymbol{b}(3)$ 的数值为向量的第 3 个元素 b_3。由于 MATLAB 的编译器能够知道矩阵 $\boldsymbol{b}$ 为一个 (1×3) 阶矩阵，因此可以不用双下标表示。当 $\boldsymbol{b}$ 为前面定义的向量时，$\boldsymbol{b}(3)$ 和 $\boldsymbol{b}(1,3)$ 是一样的。但如写成 $\boldsymbol{b}(3,1)$，则系统会给出错误信息，因为该向量并没有定义第 3 行。

相反，如果令：

$$\boldsymbol{b} = [b_1 \ b_2 \ b_3 \cdots \ b_n]'$$

则会创建一个列向量，即一个 $(n \times 1)$ 阶矩阵。如要确定向量中第 3 个元素的值，则同样可写为 $\boldsymbol{b}(3)$，MATLAB 会返回相应于 b_3 的值。这也等同于写成 $\boldsymbol{b}(3,1)$。但如果写为 $\boldsymbol{b}(1,3)$，则系统会给出错误信息，因为该向量没有定义第 3 列。

若想创建[-2,1,3,5,7,9,10]的向量 $\boldsymbol{x}$，可通过下式实现：

```
x = [-2 1:2:9 10]
```

或

```
x = [-2,1,3,5,7,9,10]
```

这意味着向量中的元素分别为 $x_1 = -2, x_2 = 1, x_3 = 3, x_4 = 5, x_5 = 7, x_6 = 9, x_7 = 10$，且向量长度 length$(x) = 7$。可通过 MATLAB 表达式 $\boldsymbol{x}(j), j = 1,2, \cdots, 7$ 访问向量 $\boldsymbol{x}$ 中的元素。例如，表达式 $\boldsymbol{x}(5)$ 返回值为 7。

当向量与标量进行加减运算时，标量与向量中的每一个元素相加减。因此：

```
z = x - 1
```

的结果为 $z = [-3 \ 0 \ 2 \ 4 \ 6 \ 8 \ 9]$。但是对于乘法、除法和幂运算，其运算规则却有严格的限制，详见第 2.6 节。

另一方面，如果仅想修改向量中的某些元素。例如，令 $z = [-2 \ 1 \ 3 \ 5 \ 7 \ 9 \ 10]$。要求向量的第 2 个元素除以 2，则可写为：

```
z=[-2 1 3 5 7 9 10];
z(2)=z(2)/2;
```

结果为 $z=[-2\ 0.5\ 3\ 5\ 7\ 9\ 10]$。如同时要求第 3 和第 4 个元素乘以 3 减去 1,则可写为:

```
z=[-2 1 3 5 7 9 10];
z(2)=z(2)/2;
z(3:4)=z(3:4)*3-1;
```

结果为 $z=[-2\ 0.5\ 8\ 14\ 7\ 9\ 10]$。注意向量中的其他元素保持不变。

此外还有几种方法可以访问向量中的元素。8 个元素的向量如下:

```
y=[-1 6 15 -7 31 2 -4 -5];
```

如要创建一个由 y 的第 3 到第 5 个元素组成的新向量 x,可写为:

```
x=y(3:5)
```

结果即创建了 3 个元素的向量 $x=[15\ \ -7\ 31]$。

如希望创建一个由 y 的前 2 个元素和后 2 个元素组成的向量 x,可输入:

```
x=[y(1) y(2) y(7) y(8)]
```

或

```
x=y([1 2 7 8])
```

此外,还可先定义一个向量,然后再按如下方法使用这个向量:

```
index=[1 2 7 8];
x=y(index)
```

后面的两种方法是非常有用的。对应于向量 y 有 8 个元素的向量 z:

```
y=[-1 6 15 -7 31 2 -4 -5];
z=[10 20 30 40 50 60 70 80];
```

用 sort 函数把向量 y 按升序排列,然后根据向量 y 中新的元素顺序重新排列向量 z。由 Help 文件可得到 sort 函数的调用格式为:

```
[ynew,indx]=sort(y)
```

其中 ***ynew*** 是重新排列后的向量 y,***indx*** 是一个由元素位置所构成的向量,它表示排列后向量 y 中元素的原始位置号。因此,程序:

```
y=[-1 6 15 -7 31 2 -4 -5];
z=[10 20 30 40 50 60 70 80];
[ynew,indx]=sort(y)
znew=z(indx)
```

的执行结果为:

```
ynew→[-7 -5 -4 -1 2 6 15 31]
indx→[4 8 7 1 6 2 3 5]
znew→[40 80 70 10 60 20 30 50]
```

可以看出，*indx*(1) = 4 表明 *ynew*(1)就是 *y*(4)。相对 *z* 而言，简单利用顺序向量 *indx*，可定义出 *znew*。

可以通过 find 命令来扩展这种功能。find 命令可找到满足用户指定条件或表达式的向量(或矩阵)元素的位置(不是元素值)。使用 find 命令在 *y* 中选出非正元素，组成一个新的向量 *s* 的过程如下。在 MATLAB 中关系运算符"< ="代表"≤"(见表 4.1)。

```
y=[-1 6 15 -7 31 2 -4 -5];
indxx=find(y<=0)
```

得出：

```
indxx→[1 4 7 8]
```

因此，

```
y=[-1 6 15 -7 31 2 -4 -5];
indxx=find(y<=0);
s=y(indxx)
```

得出：

```
s→[-1 -7 -4 -5]
```

命令行可以写为如下的紧凑形式：

```
y=[-1 6 15 -7 31 2 -4 -5];
s=y(find(y<=0))
```

find 函数的其他用法见第 6.3.8 节。

MATLAB 中隐含向量和矩阵说明的一个最大好处就是可帮助用户实现对于一组值的一系列操作。例如，如希望确定 sin(x)的值，其中 x 在 $-\pi \leq x \leq \pi$ 的区间内以 $\pi/5$ 的步长变化。则 MATLAB 语句：

```
x=-pi:pi/5:pi;
y=sin(x)
```

将生成如下向量，其元素个数 n = length(y) = 11：

```
y→[0.0000 -0.5878 -0.9511 -0.9511 -0.5878 0.0000 0.5878 0.9511 0.9511
    0.5878 0.0000]
```

2.4 矩阵的创建

对于(4×3)阶矩阵 $\boldsymbol{a}$：

$$\boldsymbol{a} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix} \rightarrow (4 \times 3)$$

可以通过以下几种方法来创建。考虑到元素 a_{11}、a_{12}和 a_{13}的组合可以创建向量 $\boldsymbol{v}_1$：

$$\boldsymbol{v}_1 = [a_{11}\ \ a_{12}\ \ a_{13}]$$

类似的方法可以创建其他 3 个向量 $\boldsymbol{v}_2$、$\boldsymbol{v}_3$、$\boldsymbol{v}_4$：

$$\boldsymbol{v}_2 = [a_{21}\ \ a_{22}\ \ a_{23}]$$

$$\boldsymbol{v}_3 = [a_{31}\ \ a_{32}\ \ a_{33}]$$

$$\boldsymbol{v}_4 = [a_{41}\ \ a_{42}\ \ a_{43}]$$

用这 4 个向量可以创建矩阵 $\boldsymbol{a}$：

$$\boldsymbol{a} = [v_1; v_2; v_3; v_4]$$

其中分号表示行的结束。每行必须具有相同的列数。这种表达式也可以用下面的方法直接创建：

$$\boldsymbol{a} = [a_{11}\ \ a_{12}\ \ a_{13}; a_{21}\ \ a_{22}\ \ a_{23}; a_{31}\ \ a_{32}\ \ a_{33}; a_{41}\ \ a_{42}\ \ a_{43}]$$

或采用更为形像的描述方法：

$$\begin{aligned} \boldsymbol{a} = [&a_{11}\ \ a_{12}\ \ a_{13}; \ldots \\ &a_{21}\ \ a_{22}\ \ a_{23}; \ldots \\ &a_{31}\ \ a_{32}\ \ a_{33}; \ldots \\ &a_{41}\ \ a_{42}\ \ a_{43}] \end{aligned}$$

其中省略号(…)是必需的，见第 1 章附录 A 表 A.1。最后，矩阵的创建还可以通过在每一行的末尾处按下 Enter 键来完成。例如：

```
a = [a11 a12 a13
    a21 a22 a23
    a31 a32 a33
    a41 a42 a43];
```

以上几种形式中，a_{ij}可以是数值、变量、表达式或字符串。如果是变量或表达式，则执行该语句前一定要首先通过用户输入或通过前面执行的表达式对变量或由变量组成的表达式进行赋值。表达式和变量可以任意组合形式出现。如果是字符串，则每一行中的字母个数应相同，详见第 3.1 节。

两个非常有用的函数可以用于生成矩阵的元素：

```
one = ones(r,c)
```

可创建一个元素为 1 的$(r \times c)$阶矩阵。

```
zero = zeros(r,c)
```

可创建一个元素为 0 的$(r \times c)$阶矩阵。这些函数可方便地代替表达式 *one*(1:r,1:c) = 1 和 *zero*(1:r,1:c) = 0 的功能。因此，

```
on = ones(2,5)
```

创建一个(2×5)阶矩阵：

```
1 1 1 1 1
1 1 1 1 1
```

以及：

```
zer = zeros(3,2)
```

创建一个(3×2)阶零矩阵：

```
0  0
0  0
0  0
```

考虑下面(3×5)阶矩阵的结构：

$$a = \begin{bmatrix} 3 & 5 & 7 & 9 & 11 \\ 20.0 & 20.25 & 20.5 & 20.75 & 21.0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \rightarrow (3\times 5)$$

它可通过如下语句创建：

```
a = [3:2:11;linspace(20,21,5);ones(1,5)]
```

创建后的矩阵如下：

```
 3.0000   5.0000   7 0000   9.0000  11.0000
20.0000  20.2500  20.5000  20.7500  21.0000
 1.0000   1.0000   1.0000   1.0000   1.0000
```

矩阵元素的访问如图 2.1 所示。因此，

```
a(1,1)→3
a(3,4)→1
a(:,2)→[5 20.25 1]'
a(2,:)→[20 20.25 20.5 20.75 21]
```

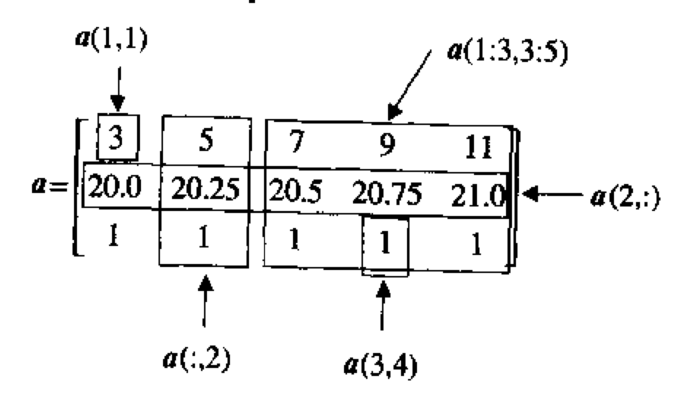


图 2.1　访问矩阵元素

且，

```
a(1:3,3:5)→[7 9 11; 20.5,20.75 21; 1 1 1]
```

是一个(3×3)阶矩阵。而，

```
a(:,2)
```

表示第 2 列中的所有元素。而，

```
a(2,:)
```

表示第 2 行中的所有元素。同时，看到书写矩阵 a 的元素时使用了冒号，这里步长为 +1。因此，

```
a(1:3,3:5)
```

意味着矩阵 a 的第 1 行到第 3 行，第 3 列到第 5 列。如果：

$$b = a(1:3,3:5) = \begin{bmatrix} 7 & 9 & 11 \\ 20.5 & 20.75 & 21 \\ 1 & 1 & 1 \end{bmatrix} \to (3 \times 3)$$

则定义了一个新的(3×3)阶矩阵 b。

创建一个大小与矩阵 a 相同但元素都等于 4 的矩阵，可由下面的表达式实现：

```
a=[3:2:11; linspace(20,21,5);ones(1,5)];
z=4*ones(size(a))
```

结果生成：

```
4 4 4 4 4
4 4 4 4 4
4 4 4 4 4
```

可以采用类似于改变向量中元素的方法改变矩阵中的元素，令：

```
z=magic(4)
```

结果显示：

```
16   2   3  13
 5  11  10   8
 9   7   6  12
 4  14  15   1
```

函数 magic 用于产生特殊矩阵，即矩阵中任意行或任意列中元素之和，以及两条对角线上的元素之和都相等。对于(4×4)阶矩阵来说其值为 34。实现第 2 行元素除以 2，第 2 列加到第 4 列的程序为：

```
z=magic(4);
z(2,:)=z(2,:)/2;
z(:,4)=z(:,4)+z(:,2);
```

结果将生成具有下列元素的矩阵 z。

```
 16    2   3   15
2.5  5.5   5  9.5
  9    7   6   19
  4   14  15   15
```

使用下面的语句将矩阵中对角元素置为 0：

```
z=magic(4);
z=z-diag(diag(z))
```

结果为:

```
0    2    3    13
5    0    10   8
9    7    0    12
4    14   15   0
```

若将对角元素的值置为 5,可以采用下面的语句:

```
z=magic(4);
z=z-diag(diag(z))+5*eye(4)
```

结果为:

```
5    2    3    13
5    5    10   8
9    7    5    12
4    14   15   5
```

若将矩阵 z 的对角元素分别置为 11,23,54,61,可以采用下面的语句:

```
z=magic(4);
z=z-diag(diag(z))+diag([11 23 54 61])
```

结果为:

```
11   2    3    13
5    23   10   8
9    7    54   12
4    14   15   61
```

MATLAB 提供了两个创建矩阵的函数,它们是通过复制标量、列向量、行向量和块矩阵来创建矩阵的。这两个函数为:

```
repmat
```

和

```
meshgrid
```

函数 repmat 的调用格式为:

```
repmat(x,r,c)
```

其中 x 可以是标量、向量或矩阵,r 是 x 的行数,c 是 x 的列数。函数 repmat 可创建任意长度的列向量和行向量,而且每一个元素都具有相同的值。创建包含 6 个元素且每个元素的值都为 45.72 的行向量 w,可输入:

```
w=repmat(45.72,1,6)
```

该表达式等同于:

```
w=[45.72, 45.72, 45.72, 45.72, 45.72, 45.72]
```

也可用下列语句创建向量 w:

```
w(1,1:6) = 45.72
```

如要创建所有值全为 45 72 的(3×3)阶矩阵,可输入:

```
w = repmat(45.72,3,3)
```

或者

```
w(1:3,1:3) = 45.72
```

每个表达式在 MATLAB 命令窗口都可生成矩阵:

```
45.7200  45.7200  45.7200
45.7200  45.7200  45.7200
45.7200  45.7200  45.7200
```

考虑向量:

$$s = [a_1\ a_2\ a_3\ a_4]$$

表达式:

```
v = repmat(s,3,1)
```

可创建数值相等[①] 的矩阵:

$$v = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ a_1 & a_2 & a_3 & a_4 \\ a_1 & a_2 & a_3 & a_4 \end{bmatrix}$$

被创建的向量 v 有 3 行,每行有 4 列。表达式:

```
repmat(s,3,2)
```

则创建了如下数值相等的矩阵:

$$v = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 & a_1 & a_2 & a_3 & a_4 \\ a_1 & a_2 & a_3 & a_4 & a_1 & a_2 & a_3 & a_4 \\ a_1 & a_2 & a_3 & a_4 & a_1 & a_2 & a_3 & a_4 \end{bmatrix}$$

另一方面,命令:

```
v = repmat(s',1,3)
```

创建每列数值等于列向量 s',共 3 列,每列 4 行的矩阵:

$$v = \begin{bmatrix} a_1 & a_1 & a_1 \\ a_2 & a_2 & a_2 \\ a_3 & a_3 & a_3 \\ a_4 & a_4 & a_4 \end{bmatrix}$$

① 所谓数值相等意味着在 MATLAB 中已为 v_v赋值。此处的表达形式是为了表明结果数组中元素的排列,并说明 repmat 函数的操作过程。

表达式:

```
repmat(s',2,3)
```

创建如下数值相等的矩阵:

$$
v=\begin{bmatrix} a_1 & a_1 & a_1 \\ a_2 & a_2 & a_2 \\ a_3 & a_3 & a_3 \\ a_4 & a_4 & a_4 \\ a_1 & a_1 & a_1 \\ a_2 & a_2 & a_2 \\ a_3 & a_3 & a_3 \\ a_4 & a_4 & a_4 \end{bmatrix}
$$

若有两个行向量 s 和 t,则 MATLAB 表达式:

```
[u,v] = meshgrid(s,t)
```

与下述的两条指令结果相同:

```
u = repmat(s,length(t),1)
v = repmat(t',1,length(s))
```

每种情况下,u 和 v 都是[length(t) × length(s)]阶矩阵。因此,如果:

$$
s=[\,s_1\ s_2\ s_3\ s_4\,]
$$

$$
t=[\,t_1\ t_2\ t_3\,]
$$

则命令:

```
[u,v] = meshgrid(s,t)
```

将生成两个(3×4)阶矩阵:

$$
u=\begin{bmatrix} s_1 & s_2 & s_3 & s_4 \\ s_1 & s_2 & s_3 & s_4 \\ s_1 & s_2 & s_3 & s_4 \end{bmatrix} \tag{2.1a}
$$

$$
v=\begin{bmatrix} t_1 & t_1 & t_1 & t_1 \\ t_2 & t_2 & t_2 & t_2 \\ t_3 & t_3 & t_3 & t_3 \end{bmatrix} \tag{2.1b}
$$

函数 meshgrid 也可以用来仅返回一个矩阵,如:

```
w = meshgrid(s,t)
```

生成 $w=u$,其中 u 由式(2.1a)给出。例 2.2 给出了说明。

矩阵操作函数 fliplr(a)和 flipud(a)在某些应用中非常有用,它们可实现对矩阵行和列的旋转操作。考虑(2×5)阶矩阵:

$$
a=\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \end{bmatrix} \rightarrow (2\times 5)
$$

用如下语句创建：

$$\boldsymbol{a}=[a_{11}\ a_{12}\ a_{13}\ a_{14}\ a_{15};a_{21}\ a_{22}\ a_{23}\ a_{24}\ a_{25}]$$

那么，

$$\text{fliplr}(a)=\begin{bmatrix} a_{15} & a_{14} & a_{13} & a_{12} & a_{11} \\ a_{25} & a_{24} & a_{23} & a_{22} & a_{21} \end{bmatrix}\rightarrow(2\times5)$$

$$\text{flipud}(a)=\begin{bmatrix} a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \end{bmatrix}\rightarrow(2\times5)$$

和

$$\text{flipud}(\text{fliplr}(a))=\begin{bmatrix} a_{25} & a_{24} & a_{23} & a_{22} & a_{21} \\ a_{15} & a_{14} & a_{13} & a_{12} & a_{11} \end{bmatrix}\rightarrow(2\times5)$$

函数 fliplr(a)与 flipud(a)所产生的结果同样可由矩阵下标中的冒号得到，例如：

```
c = fliplr(a)
```

产生与下式相同的结果：

```
c = a(:,length(a):-1:1)
```

考虑向量：

$$\boldsymbol{c}=[a\ \text{fliplr}(a)]'=\begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \\ a_{13} & a_{23} \\ a_{14} & a_{24} \\ a_{15} & a_{25} \\ a_{15} & a_{25} \\ a_{14} & a_{24} \\ a_{13} & a_{23} \\ a_{12} & a_{22} \\ a_{11} & a_{21} \end{bmatrix}\rightarrow(10\times2)$$

产生的第 5 行和第 6 行相同。如想去掉其中的一行，可以通过表达式[]使其中一行为空，其中方括号中没有空格。用表达式：

```
c(length(a),:) = []
```

或

```
c(length(a)+1,:) = []
```

可生成矩阵：

$$c=\begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \\ a_{13} & a_{23} \\ a_{14} & a_{24} \\ a_{15} & a_{25} \\ a_{14} & a_{24} \\ a_{13} & a_{23} \\ a_{12} & a_{22} \\ a_{11} & a_{21} \end{bmatrix} \rightarrow (9\times 2)$$

其中 c 为(9×2)阶矩阵。表达式 c(length(a),:)=[]表示矩阵 c 中第 length(a)行中所有元素的值被指定为[](这种情况下,相当于值被删除)。尽管知道矩阵 a 的长度为 5,但是作为训练,还是使用函数 length(a)让 MATLAB 计算为好。

通过 MATLAB 三种不同操作的结果来进一步说明上述用法。首先创建两个(2×5)阶矩阵 a 和 b。

$$a=\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \end{bmatrix} \quad b=\begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} \end{bmatrix}$$

考虑它们在三种 MATLAB 操作中的用法:

相加:$c=a+b$

$$c=\begin{bmatrix} a_{11}+b_{11} & a_{12}+b_{12} & a_{13}+b_{13} & a_{14}+b_{14} & a_{15}+b_{15} \\ a_{21}+b_{21} & a_{22}+b_{22} & a_{23}+b_{23} & a_{24}+b_{24} & a_{25}+b_{25} \end{bmatrix} \rightarrow (2\times 5)$$

c 为(2×5)阶矩阵。

列扩展:$c=[a\ b]$

$$c=\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & b_{11} & b_{12} & b_{13} & b_{14} & b_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & b_{21} & b_{22} & b_{23} & b_{24} & b_{25} \end{bmatrix} \rightarrow (2\times 10)$$

c 为(2×10)阶矩阵。

行扩展:$c=[a;b]$

$$c=\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ b_{11} & b_{12} & b_{13} & b_{14} & b_{15} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} \end{bmatrix} \rightarrow (4\times 5)$$

c 为(4×5)阶矩阵。

更进一步,如果,

$$x=[x_1\ x_2\ x_3]$$
$$y=[y_1\ y_2\ y_3]$$

那么,

```
z=[x' y']
```

或

```
z=[x;y]'
```

可生成:

$$z=\begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{bmatrix} \rightarrow (2\times 3)$$

然而,

```
z=[x';y']
```

可生成:

$$z=\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} \rightarrow (6\times 1)$$

在按某种顺序存放数据时,这两种变换是非常有用的。

2.5 点运算

本节介绍 MATLAB 中的点(.)运算。在 MATLAB 中点运算是对同阶矩阵中逐个元素进行的算术运算。考虑下面的(3×4)阶矩阵:

$$\boldsymbol{x}=\begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \end{bmatrix}$$

和

$$\boldsymbol{m}=\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix}$$

直接写出 MATLAB 点运算:

$$z_m=\boldsymbol{x}.*\boldsymbol{m}=\begin{bmatrix} x_{11}*m_{11} & x_{12}*m_{12} & x_{13}*m_{13} & x_{14}*m_{14} \\ x_{21}*m_{21} & x_{22}*m_{22} & x_{23}*m_{23} & x_{24}*m_{24} \\ x_{31}*m_{31} & x_{32}*m_{32} & x_{33}*m_{33} & x_{34}*m_{34} \end{bmatrix} \tag{2.2a}$$

$$z_d=\boldsymbol{x}./\boldsymbol{m}=\begin{bmatrix} x_{11}/m_{11} & x_{12}/m_{12} & x_{13}/m_{13} & x_{14}/m_{14} \\ x_{21}/m_{21} & x_{22}/m_{22} & x_{23}/m_{23} & x_{24}/m_{24} \\ x_{31}/m_{31} & x_{32}/m_{32} & x_{33}/m_{33} & x_{34}/m_{34} \end{bmatrix} \tag{2.2b}$$

$$z_e=\boldsymbol{x}.\hat{}\boldsymbol{m}=\begin{bmatrix} x_{11}\hat{}m_{11} & x_{12}\hat{}m_{12} & x_{13}\hat{}m_{13} & x_{14}\hat{}m_{14} \\ x_{21}\hat{}m_{21} & x_{22}\hat{}m_{22} & x_{23}\hat{}m_{23} & x_{24}\hat{}m_{24} \\ x_{31}\hat{}m_{31} & x_{32}\hat{}m_{32} & x_{33}\hat{}m_{33} & x_{34}\hat{}m_{34} \end{bmatrix} \tag{2.2c}$$

也可以表示为：

$$z_{mij} = x_{ij}m_{ij} \quad \{或\ \boldsymbol{zm}(i,j) = \boldsymbol{x}(i,j) * \boldsymbol{m}(i,j)\}$$
$$z_{dij} = x_{ij}/m_{ij} \quad \{或\ \boldsymbol{zd}(i,j) = \boldsymbol{x}(i,j)/\boldsymbol{m}(i,j)\}$$
$$z_{eij} = x_{ij}^{m_{ij}} \quad \{或\ \boldsymbol{ze}(i,j) = \boldsymbol{x}(i,j)\hat{}\ \boldsymbol{m}(i,j)\}$$

其中 $i=1,2,3$ 且 $j=1,2,3,4$。注意点运算符号必须要放在乘号、除号和指数运算符号前面。对于矩阵的加法和减法来说，点运算是没必要的，见式(2.6)。

对于点乘来说，如果 $x = x_0$，其中 x_0 为标量常数，则点乘和乘法运算是一样的。

$$x_0 .* \boldsymbol{m} = \begin{bmatrix} x_0 * m_{11} & x_0 * m_{12} & x_0 * m_{13} & x_0 * m_{14} \\ x_0 * m_{21} & x_0 * m_{22} & x_0 * m_{23} & x_0 * m_{24} \\ x_0 * m_{31} & x_0 * m_{32} & x_0 * m_{33} & x_0 * m_{34} \end{bmatrix} = x_0 \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} = x_0 * \boldsymbol{m}$$

类似，当 $\boldsymbol{m} = m_0$ = 标量常数时，则：

$$\boldsymbol{x} .* \boldsymbol{m} = \boldsymbol{x} * m_0$$

对于点除运算来说，当 $\boldsymbol{m} = m_0$ 且 m_0 为标量常数时，为：

$$\boldsymbol{x}./m_0 = \begin{bmatrix} x_{11}/m_0 & x_{12}/m_0 & x_{13}/m_0 & x_{14}/m_0 \\ x_{21}/m_0 & x_{22}/m_0 & x_{23}/m_0 & x_{24}/m_0 \\ x_{31}/m_0 & x_{32}/m_0 & x_{33}/m_0 & x_{34}/m_0 \end{bmatrix}$$
$$= \frac{1}{m_0}\begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \end{bmatrix} = \boldsymbol{x}/m_0$$

此时点运算是不必要的。然而，当 $\boldsymbol{x} = x_0$ 为标量常数时，为：

$$x_0./\boldsymbol{m} = \begin{bmatrix} x_0/m_{11} & x_0/m_{12} & x_0/m_{13} & x_0/m_{14} \\ x_0/m_{21} & x_0/m_{22} & x_0/m_{23} & x_0/m_{24} \\ x_0/m_{31} & x_0/m_{32} & x_0/m_{33} & x_0/m_{34} \end{bmatrix}$$
$$= x_0\begin{bmatrix} 1/m_{11} & 1/m_{12} & 1/m_{13} & 1/m_{14} \\ 1/m_{21} & 1/m_{22} & 1/m_{23} & 1/m_{24} \\ 1/m_{31} & 1/m_{32} & 1/m_{33} & 1/m_{34} \end{bmatrix} = x_0/\boldsymbol{m}$$

这里的点运算是必需的。

很显然，当 $\boldsymbol{m} = m_0$ 为标量常数或 $\boldsymbol{x} = x_0$ 为标量常数时，指数运算总要用到点运算。因此，如果 $\boldsymbol{z} = \boldsymbol{x}.\hat{}\boldsymbol{m}$，则 x_0 为标量常数时：

$$\boldsymbol{z} = x_0.\hat{}\boldsymbol{m}$$

或者，m_0 为标量常数时：

$$\boldsymbol{z} = \boldsymbol{x}.\hat{}m_0$$

为说明指数的点运算，考虑计算 2^j，其中 $j = 1,2,\cdots,8$。程序为：

```
x = 1:8;
y = 2.^x
```

结果为：

```
2 4 8 16 32 64 128 256
```

因此，在 MATLAB 中，放在指数操作符(^)前面的小数点意味着以 2 为标量，在矩阵 $\boldsymbol{x}$ 的每个取值处计算其指数值，并将计算结果作为向量 $\boldsymbol{y}$ 的元素值。上面的程序可简写为：

```
y=2.^(1:8) % 或 y=2.^[1:8]
```

反之，如求取 x^2 的值，则程序为：

```
y=(1:8).^2 % 或 y=[1:8].^2
```

结果为：

```
1 4 9 16 25 36 49 64
```

如果令 $f(y)$代替任意一个函数，如 sin、cos、cosh、贝塞尔函数等，则当矩阵 $\boldsymbol{y}$ 为一个(3×4)阶矩阵时：

$$z=f(y)=\begin{bmatrix} f(y_{11}) & f(y_{12}) & f(y_{13}) & f(y_{14}) \\ f(y_{21}) & f(y_{22}) & f(y_{23}) & f(y_{24}) \\ f(y_{31}) & f(y_{32}) & f(y_{33}) & f(y_{34}) \end{bmatrix}$$

在函数阶数相同的情况下，可实现与点运算的联合运算。例如，如果 $\boldsymbol{a}$、$\boldsymbol{b}$、$\boldsymbol{c}$、$\boldsymbol{d}$ 和 $\boldsymbol{f}$ 都是(3×2)阶矩阵，则表达式：

$$\boldsymbol{z}=\left[\tan\boldsymbol{a}-\boldsymbol{f}\left(\frac{\boldsymbol{b}}{\boldsymbol{c}}\right)^{\boldsymbol{d}}\right]^2$$

可写为：

```
z=(tan(a)-f.*(b./c).^d).^2;
```

$\boldsymbol{z}$ 中各元素的数值可通过以下的表达式计算得出：

$$z=\begin{bmatrix} (\tan(a_{11})-f_{11}*(b_{11}/c_{11})\hat{}d_{11})\hat{}2 & (\tan(a_{12})-f_{12}*(b_{12}/c_{12})\hat{}d_{12})\hat{}2 \\ (\tan(a_{21})-f_{21}*(b_{21}/c_{21})\hat{}d_{21})\hat{}2 & (\tan(a_{22})-f_{22}*(b_{22}/c_{22})\hat{}d_{22})\hat{}2 \\ (\tan(a_{31})-f_{31}*(b_{31}/c_{31})\hat{}d_{31})\hat{}2 & (\tan(a_{32})-f_{32}*(b_{32}/c_{32})\hat{}d_{32})\hat{}2 \end{bmatrix}$$

点运算的另一个应用是函数 meshgrid。利用下述语句：

```
[u,v]=meshgrid(s,t)
```

通过向量 $\boldsymbol{s}=[s_1\ s_2\ s_3\ s_4]$和 $\boldsymbol{t}=[t_1\ t_2\ t_3]$可生成两个(3×4)阶矩阵：

$$\boldsymbol{u}=\begin{bmatrix} s_1 & s_2 & s_3 & s_4 \\ s_1 & s_2 & s_3 & s_4 \\ s_1 & s_2 & s_3 & s_4 \end{bmatrix} \quad 和 \quad \boldsymbol{v}=\begin{bmatrix} t_1 & t_1 & t_1 & t_1 \\ t_2 & t_2 & t_2 & t_2 \\ t_3 & t_3 & t_3 & t_3 \end{bmatrix} \tag{2.3}$$

若让 $\boldsymbol{u}$ 和 $\boldsymbol{v}$ 中的相关元素相乘，则点乘：

```
z=u.*v
```

可生成如下结果[参见式(2.2a)]：

$$z=\begin{bmatrix} s_1*t_1 & s_2*t_1 & s_3*t_1 & s_4*t_1 \\ s_1*t_2 & s_2*t_2 & s_3*t_2 & s_4*t_2 \\ s_1*t_3 & s_2*t_3 & s_3*t_3 & s_4*t_3 \end{bmatrix} \tag{2.4}$$

z 中的元素是向量 s 和 t 中元素的所有可能乘积值。当进行加、减、除和指数运算时，结果相似。只不过是将乘法符号(*)用其他运算符所替代而已。

检验：

```
sum
```

和

```
cumsum
```

函数，它们经常用于点操作之中。首先看 sum 函数。当 sum 函数的参数为向量时，sum 函数将计算向量中所有元素之和，并返回此值。当参数为矩阵时，函数将矩阵中的元素按列相加，并返回一个长度等于原始矩阵列数的行向量。因此，如果 z 是一个(3×4)阶矩阵且其元素为 z_{ij}，则：

$$\mathrm{sum}(z) = \left[\sum_{n=1}^{3} z_{n1} \quad \sum_{n=1}^{3} z_{n2} \quad \sum_{n=1}^{3} z_{n3} \quad \sum_{n=1}^{3} z_{n4}\right] \rightarrow (1 \times 4) \tag{2.5a}$$

是一个 4 元素向量。而：

$$\mathrm{sum}(z') = \left[\sum_{n=1}^{4} z_{1n} \quad \sum_{n=1}^{4} z_{2n} \quad \sum_{n=1}^{4} z_{3n}\right] \rightarrow (1 \times 3) \tag{2.5b}$$

是一个 3 元素向量。z 可以是上述任何一种点运算的结果，或者是组合点运算的结果。

向量 v 有 n 个元素 v_j，cumsum 函数对其操作生成另一个长度为 n 的向量，其元素为：

$$y = \mathrm{cumsum}(v) = \left[\sum_{k=1}^{1} v_k \quad \sum_{k=1}^{2} v_k \cdots \sum_{k=1}^{n} v_k\right] \rightarrow (1 \times n)$$

如果 w 是由元素 w_{jk} 组成的一个($m \times n$)阶矩阵，则 cumsum(w)为如下矩阵：

$$y = \mathrm{cumsum}(w) = \begin{bmatrix} \sum_{k=1}^{1} w_{k1} & \sum_{k=1}^{1} w_{k2} & \cdots & \sum_{k=1}^{1} w_{kn} \\ \sum_{k=1}^{2} w_{k1} & \sum_{k=1}^{2} w_{k2} & & \\ \vdots & & \ddots & \\ \sum_{k=1}^{m} w_{k1} & & & \sum_{k=1}^{m} w_{kn} \end{bmatrix} \rightarrow (m \times n)$$

为说明 sum 函数的应用，参见如下公式：

$$z = \sum_{m=1}^{4} m^m$$

计算表达式的程序为：

```
m = 1:4;
z = sum(m.^m)
```

计算结果为 $z = 288$。程序可简化为：

```
z = sum((1:4).^(1:4))
```

通过计算下式① 对本节的内容做一小结,式中 $N=305$,且 x 在 $0 \leqslant x \leqslant 2$ 的区间内按 5 等分取值。

$$\operatorname{sech} x = 4\pi \sum_{n=1,3,5}^{N \to \infty} \frac{n(-1)^{(n-1)/2}}{(n\pi)^2 + 4x^2}$$

并将计算值与其理论值做一比较。程序为:

```
nn = 1:2:305;                                                    %(1×153)
xx = linspace(0,2,5);                                            %(1×5)
[x,n] = meshgrid(xx,nn);                                         %(153×5)
s = 4 * pi * sum(n. * (-1).^((n-1)/2) /((pi * n).^2 + 4 * x.^2));  %(1×5)
se = sech(xx);                                                   %(1×5)
compare = [s' se']                                               %(5×2)
```

程序运行后,在 MATLAB 命令窗口中显示结果:

```
1.0021  1.0000
0.8869  0.8868
0.6501  0.6481
0.4272  0.4251
0.2679  0.2658
```

由于 sum 函数可实现行元素的按列求和,所以选定 meshgrid 函数的参数顺序,生成(153×5)阶矩阵。程序简化为:

```
[x,n] = meshgrid(linspace(0,2,5),1:2:305);
s = 4 * pi * sum(n. * (-1).^((n-1)/2)./((pi * n).^2 + 4 * x.^2));
compare = [s' sech(linspace(0,2,5)')]
```

2.6 矩阵的数学运算

在本节将介绍矩阵的几种运算:加、减、乘、转置、行列式、方程式及其根(特征值)。这些运算经常用来求解各种工程问题。

2.6.1 加法和减法

如有两个$(m \times n)$阶矩阵 $\boldsymbol{a}$ 和 $\boldsymbol{b}$,则有:

$$\boldsymbol{a} \pm \boldsymbol{b} = \begin{bmatrix} a_{11} \pm b_{11} & a_{12} \pm b_{12} & \cdots & a_{1n} \pm b_{1n} \\ a_{21} \pm b_{21} & a_{22} \pm b_{22} & & \\ \vdots & & \ddots & \\ a_{m1} \pm b_{m1} & & & a_{mn} \pm b_{mn} \end{bmatrix} \to (m \times n \qquad (2.6)$$

2.6.2 乘法

如有$(m \times k)$阶矩阵 $\boldsymbol{a}$ 和$(k \times n)$阶矩阵 $\boldsymbol{b}$,则:

---

① 参见 L B W Jolley, *Summation of Series*, 第二版, Dover Publications, New York

$$c = ab = \begin{bmatrix} \sum_{j=1}^{k} a_{1j}b_{j1} & \sum_{j=1}^{k} a_{1j}b_{j2} & \cdots & \sum_{j=1}^{k} a_{1j}b_{jn} \\ \sum_{j=1}^{k} a_{2j}b_{j1} & \sum_{j=1}^{k} a_{2j}b_{j2} & & \vdots \\ \vdots & & \ddots & \\ \sum_{j=1}^{k} a_{mj}b_{j1} & \cdots & & \sum_{j=1}^{k} a_{mj}b_{jn} \end{bmatrix} \rightarrow (m \times n) \tag{2.7}$$

其中 c 为$(m \times n)$阶矩阵。应注意到只有当两个矩阵的相邻阶数相等时(本例中为 k),矩阵乘积的定义才成立。换句话说,只能是$(m \times k)(k \times n) = (m \times n)$。式(2.7)表明是对 k 项进行了求和操作。MATLAB 中矩阵乘积的表达式为:

```
c=a*b
```

式(2.7)矩阵乘积的扩展运算为 $f = cd = abd$,由例 2.5 中的式(2.15)给出。

通常情况下 $ab \neq ba$。但当 $m = n$ 时,且 $c = ab$,则:

$$c' = (ab)' = b'a'$$

同样,如果 a 为单位矩阵($a = I$)且 $m = n$,则:

$$Ib = bI = b$$

考虑下列级数[①]:

$$w(x,y) = \sum_{j=1}^{k} d_j e_j(x) g_j(y) = \sum_{j=1}^{k} f_j(x) g_j(y)$$

x,y 具有一系列的值,设 $x = x_1, x_2, \cdots, x_m$ 而 $y = y_1, y_2, \cdots, y_n$。$w(x,y)$的值表示为:

$$w(x_i, y_j) = \sum_{l=1}^{k} f_l(x_i) g_l(y_j) \quad i = 1,2,\cdots,m \quad j = 1,2,\cdots,n$$

它可作为$(m \times n)$阶矩阵 w 的一个元素。令 f 为$(m \times k)$阶矩阵:

$$f = \begin{bmatrix} f_1(x_1) & f_2(x_1) & \cdots & f_k(x_1) \\ f_1(x_2) & f_2(x_2) & & \\ \vdots & & \ddots & \\ f_1(x_m) & \cdots & & f_k(x_m) \end{bmatrix} \rightarrow (m \times k)$$

且 g 为$(k \times n)$阶矩阵:

$$f = \begin{bmatrix} g_1(y_1) & g_1(y_2) & \cdots & g_1(y_n) \\ g_2(y_1) & g_2(y_2) & & \\ \vdots & & \ddots & \\ g_k(y_1) & \cdots & & g_k(y_n) \end{bmatrix} \rightarrow (k \times n)$$

由式(2.7)可得出:

① 级数形式源于给定边界条件下微分方程式的解集。

$$
w = fg = \begin{bmatrix} \sum_{j=1}^{k} f_j(x_1)g_j(y_1) & \sum_{j=1}^{k} f_j(x_1)g_j(y_2) & \cdots & \sum_{j=1}^{k} f_j(x_1)g_j(y_n) \\ \sum_{j=1}^{k} f_j(x_2)g_j(y_1) & \sum_{k=1}^{k} f_j(x_2)g_j(y_2) & & \vdots \\ \vdots & & \ddots & \\ \sum_{j=1}^{k} f_j(x_m)g_j(y_1) & \cdots & & \sum_{j=1}^{k} f_j(x_m)g_j(y_n) \end{bmatrix} \rightarrow (m \times n) \tag{2.8}
$$

对于 x 和 y 的任意一种数值组合，矩阵的乘积运算都可实现级数的求和。可以看出，对于向量 x 和 y 的任何元素组合，矩阵乘积运算为级数求和提供了一种非常简便的方法。

考虑矩阵乘积的三种特殊情况：

1. 行向量和列向量乘积。
2. 列向量和行向量乘积。
3. 行向量和矩阵的乘积。

在这三种情况的讨论中，提供了一种解决工程问题的方法，这种方法充分利用 MATLAB 的优点，如简洁的表达形式和矩阵的多种运算方法。

第 1 种情况：行向量和列向量的乘积。令 a 为行向量：

$$a = [a_1\ a_2 \cdots\ a_k] \rightarrow (1 \times k)$$

其阶数为 $(1 \times k)$，b 为列向量：

$$b = [b_1\ b_2 \cdots\ b_k]' \rightarrow (k \times 1)$$

其阶数为 $(k \times 1)$。则 a 与 b 相乘的结果为标量：

$$
d = ab = [a_1 \quad a_2 \quad \cdots \quad a_k] \begin{bmatrix} b_1 \\ b_1 \\ \vdots \\ b_k \end{bmatrix} = \left[\sum_{j=1}^{k} a_j b_j \right] = \sum_{j=1}^{k} a_j b_j \rightarrow (1 \times 1) \tag{2.9}
$$

乘积的阶数为 $(1 \times k)(k \times 1) = (1 \times 1)$。这种运算称为两个向量的点乘。在 MATLAB 中上述两向量矩阵相乘的表达式定义为：

```
d = a * b
```

或

```
d = dot(a,b)
```

如果 c 是一个 $(n \times n)$ 阶矩阵，x 是一个 $(n \times 1)$ 阶列向量，则：

$$
f = x'cx = \sum_{i=1}^{n} \sum_{j=1}^{n} x_i c_{ij} x_j \tag{2.10}
$$

由于其阶数为 $(1 \times n)(n \times n)(n \times 1) = (1 \times 1)$，即结果也为标量值。式(2.10)的 MATLAB 表达式为：

```
f = x' * c * x
```

这种组合运算在求解特征值问题时极为有用(见第 9 章)。

第 2 种情况:列向量和行向量的乘积。令 $\boldsymbol{b}$ 为$(m\times 1)$阶列向量,$\boldsymbol{a}$ 为$(1\times n)$阶行向量,则乘积 $\boldsymbol{h}=\boldsymbol{ba}$ 为:

$$\boldsymbol{h}=\boldsymbol{ba}=\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}\begin{bmatrix} a_1 & a_2 \cdots & a_n \end{bmatrix}=\begin{bmatrix} b_1a_1 & b_1a_2 & \cdots & b_1a_n \\ b_2a_1 & b_2a_2 & & \\ \vdots & & \ddots & \\ b_ma_1 & & \cdots & b_ma_n \end{bmatrix}\to(m\times n) \tag{2.11}$$

结果为$(m\times n)$阶矩阵,乘积的阶数由$(m\times 1)(1\times n)=(m\times n)$给出。因此,矩阵 $\boldsymbol{h}$ 中的元素$(h_{ij}=b_ia_j)$即为矩阵 $\boldsymbol{b}$ 和矩阵 $\boldsymbol{a}$ 中所有元素乘积的组合。

作为应用的例子,讨论图 2.2 所给出的极坐标到笛卡儿坐标的变换。即:

$$x=r\cos(\theta)$$
$$y=r\sin(\theta)$$

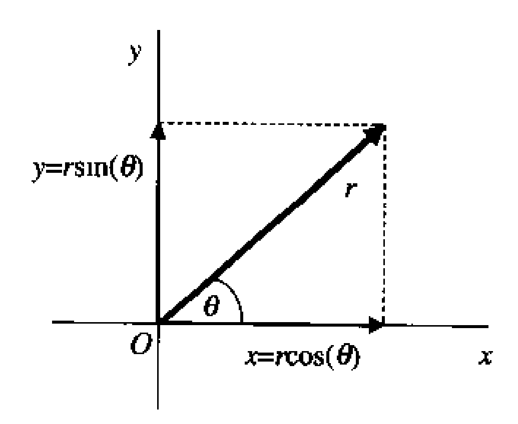


图 2.2　极坐标到笛卡儿坐标的转换

如有半径值向量 $\boldsymbol{r}=[r_1\ r_2\cdots\ r_m]$和角度值向量 $\boldsymbol{\theta}=[\theta_1\theta_2\cdots\ \theta_n]$,则相应的笛卡儿坐标值为:①

$$\boldsymbol{x}=\boldsymbol{r}'*\cos(\boldsymbol{\theta})=\begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{bmatrix}\begin{bmatrix} \cos\theta_1 & \cos\theta_2 & \cdots & \cos\theta_n \end{bmatrix}$$

$$=\begin{bmatrix} r_1\cos\theta_1 & r_1\cos\theta_2 & \cdots & r_1\cos\theta_n \\ r_2\cos\theta_1 & r_2\cos\theta_2 & & \\ \vdots & & \ddots & \\ r_m\cos\theta_1 & & \cdots & r_m\cos\theta_n \end{bmatrix} \tag{2.12a}$$

和

① 这种转换可由函数 pol2cart 实现,但 pol2cart 函数仅适用于 $m=n$ 的情况。

$$y = r' * \sin(\theta) = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{bmatrix} [\sin\theta_1 \quad \sin\theta_2 \quad \cdots \quad \sin\theta_n]$$

$$= \begin{bmatrix} r_1\sin\theta_1 & r_1\sin\theta_2 & \cdots & r_1\sin\theta_n \\ r_2\sin\theta_1 & r_2\sin\theta_2 & & \\ \vdots & & \ddots & \\ r_m\sin\theta_1 & & \cdots & r_m\sin\theta_n \end{bmatrix} \quad (2.12b)$$

这样就把极坐标映射到笛卡儿坐标中去。在下例中将说明这种变换对于绘制效果图是极为有用的。

例2.1 环型薄膜的形状

考虑固态环型薄膜的形状模型，沿外边界受压，外边界 $r = 1$：

$$z(r,\phi) = J_1(3.831\,6r)\cos(\phi)$$

其中 $J_1(x)$为第一类1阶贝塞尔函数[①]，$(r,\ \phi)$为薄膜上任意一点的极坐标值。贝塞尔函数由下式确定。

```
besselj(n,x)
```

其中 n 为阶数，x 为参数。薄膜中心处为坐标原点，薄膜的固有频率参数为3.831 6。该模型外形可由下面的外形绘制函数画出：

```
mesh(x,y,z)
```

其中(x,y)为表面函数 $z(x,y)$上一个点的平面坐标。mesh 函数将在第7.2节详细讨论。以 $\Delta r = 0.05$ 和 $\theta = \pi/20$ 为增量绘制表面图，程序如下：

```
r=[0:0.05:1]';                      %(21×1)
phi=0:pi/20:2*pi;                   %(1×41)
x=r*cos(phi);                       %(21×41)
y=r*sin(phi);                       %(21×41)
z=besselj(1,3.8316*r)cos(phi);      %(21×41)
mesh(x,y,z)
```

其中，***phi*** 为一个(1×41)阶矩阵，***r*** 为一个(21×1)阶矩阵。由式(2.12)可得出 ***x***，***y***，z 均为(21×41)阶矩阵。因为要在笛卡儿坐标系中绘出外表面图，故需进行坐标变换。由于cos、sin和贝塞尔函数允许向量作为输入参数，并且返回相同阶次的向量，所以这个过程是可实现的。程序执行的结果如图2.3所示。

---

① 参见F B Hildebrand, *Advanced Calculus for Applications*, Prentice-Hall, Saddle River, NJ, 1976年。

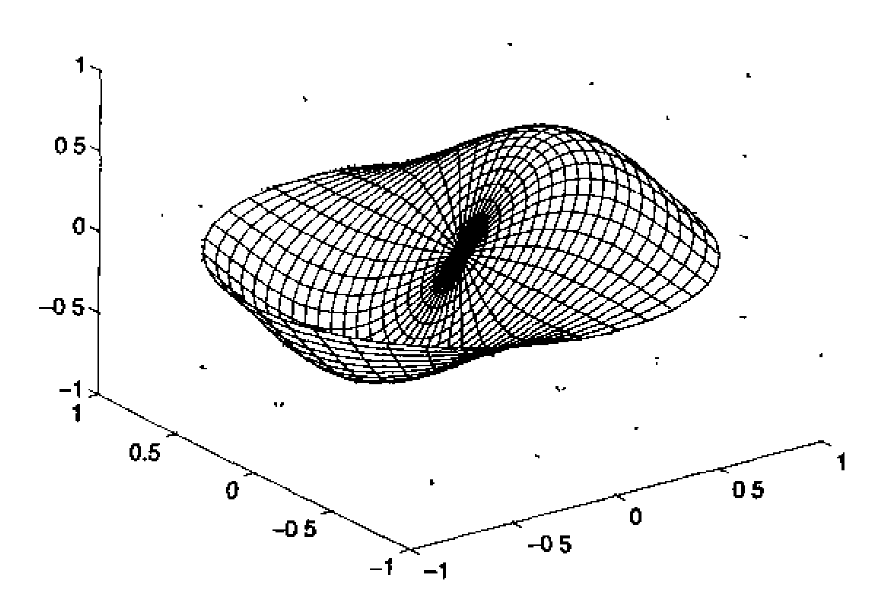


图 2.3　一个受压的固态环形薄膜外形

例 2.2　拉普拉斯方程求解

在 $u(0,\eta)=u(1,\eta)=u(\xi,1)=0$ 和 $u(\xi,0)=\xi(1-\xi)$ 边界条件下，拉普拉斯方程的解为：

$$u(\xi,\eta)=4\sum_{n=1}^{N\to\infty}\frac{1-\cos n\pi}{(n\pi)^3}e^{-n\pi\xi}\sin n\pi\eta$$

其中 $0\leqslant\eta\leqslant1,\xi\geqslant0$。若 $N=25$，增量 $\Delta\eta=0.025,\Delta\xi=0.05$ 直到 $\xi_{max}=0.7$，利用函数 mesh$(\eta,\xi,u(\xi,\eta))$ 绘制 $u(\xi,\eta)$ 表面的程序如下：

```
n = (1:25) * pi;                            %(1 x 25)
eta = 0:0.025:1;                            %(1 x 41)
xi = 0:0.05:0.7;                            %(1 x 15)
tempc = meshgrid((1 - cos(n))./n.^3,xi);    %(15 x 25)
tempe = exp( - n' * xi)';                   %(15 x 25)
tempec = tempc. * tempe;                    %(15 x 25)
temps = sin(n' * eta);                      %(25 x 41)
z = 4 * tempec * temps;                     %(15 x 41)
mesh(eta,xi,z)
```

结果如图 2.4 所示。如式(2.8)所示，矩阵乘积 *tempec* * *temps* 在 *n* 的取值范围内对 *eta* 和 *xi* 的所有组合所构成的级数进行了求和。mesh 命令中，MATLAB 允许其前两个参数 ***eta*** 和 ***xi*** 为向量，向量长度与矩阵 *z* 的阶数相统一。mesh 函数的使用方法可参见帮助文件。因为 MATLAB 表达式遵循从左到右的计算规则，故上述程序可简化为：

```
n = (1:25) * pi;
eta = 0:0.025:1;
xi = 0:0.05:0.7;
z = 4 * meshgrid((1 - cos(n))./n.^3,xi). * exp( - n' * xi)' * sin(n' * eta);
mesh(eta,xi,z)
```

第 3 种情况:行向量和矩阵的乘积。令 $\boldsymbol{b}$ 为$(m\times n)$阶矩阵且 $\boldsymbol{a}$ 为$(1\times m)$阶行向量。则乘积 $\boldsymbol{g}=\boldsymbol{ab}$ 为:

$$\boldsymbol{g}=\boldsymbol{ab}=[a_1\ a_2\ \cdots\ a_m]\begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & & \\ \vdots & & \ddots & \\ b_{m1} & & \cdots & b_{mn}\end{bmatrix}$$

$$=\left[\sum_{k=1}^{m}a_kb_{k1}\quad \sum_{k=1}^{m}a_kb_{k2}\quad\cdots\quad\sum_{k=1}^{m}a_kb_{kn}\right]\to(1\times n) \tag{2.13}$$

因为阶数乘积为$(1\times m)(m\times n)=(1\times n)$,即结果为$(1\times n)$阶行向量。

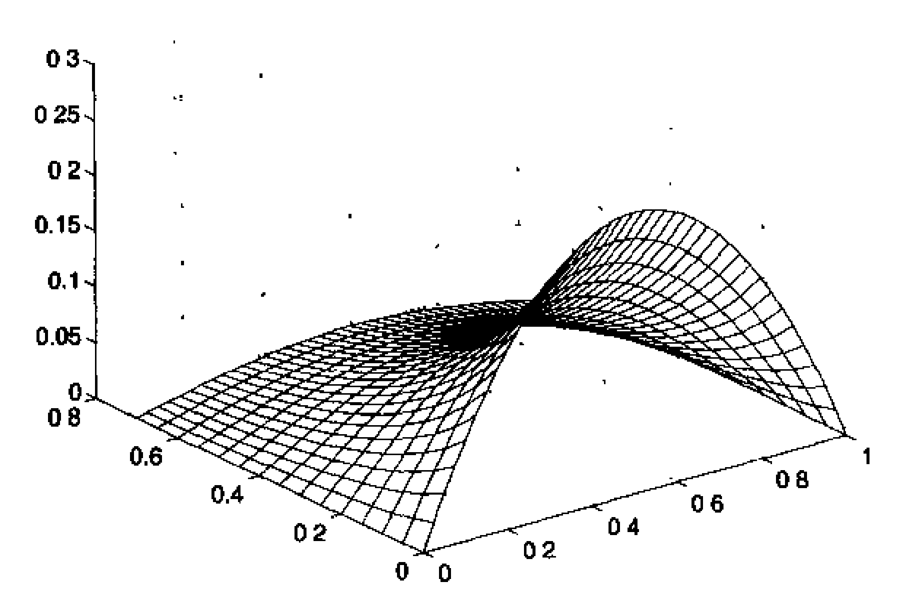


图 2.4 显示拉普拉斯方程的解

考虑如下级数:[①]

$$r(x)=\sum_{j=1}^{m}p_jh_j(x) \tag{2.14}$$

x 的取值可以是 $x_1,x_2,\cdots,x_n$。则 $r(x_i)$的表达式如下:

$$r(x_i)=\sum_{j=1}^{m}p_jh_j(x_i)\quad i=1,2,\cdots,n$$

$r(x_i)$作为$(1\times n)$阶向量 $\boldsymbol{r}$ 的一个元素。令 $\boldsymbol{p}$ 为$(1\times m)$阶向量,其元素为 p_j,$\boldsymbol{v}$ 为$(m\times n)$阶矩阵,形式如下:

$$\boldsymbol{v}=\begin{bmatrix} h_1(x_1) & h_1(x_2) & \cdots & h_1(x_n) \\ h_2(x_1) & h_2(x_2) & & \\ \vdots & & \ddots & \\ h_m(x_1) & & & h_m(x_n)\end{bmatrix}\to(m\times n)$$

① 该形式级数可由特定边界条件下微分方程的解或由周期函数的傅里叶级数展开式获得。

则 $\boldsymbol{r}=\boldsymbol{p}\boldsymbol{v}$ 可表示为：

$$\boldsymbol{r}=\boldsymbol{p}\boldsymbol{v}=\left[\sum_{j=1}^{m}p_jh_j(x_1)\quad\sum_{j=1}^{m}p_jh_j(x_2)\quad\cdots\quad\sum_{j=1}^{m}p_jh_j(x_n)\right]\rightarrow(1\times n)$$

下面的实例给出了该结论在应用中的几点说明。

例 2.3 傅里叶级数求和

脉冲宽度为 d，周期为 T 的矩形脉冲的傅里叶级数由下式给出[①]：

$$f(\tau)=\frac{d}{T}\left[1+2\sum_{n=1}^{\infty}\frac{\sin(n\pi d/T)}{n\pi\quad d/T}\cos(2\pi n\tau)\right]$$

其中 $\tau=t/T$。可以看出，本式是式(2.14)的一个具体表示形式。

对 $f(\tau)$ 的 150 个值求和并绘制当 $d/T=0.25$，$-1/2<\tau<1/2$ 时的图形。图形的绘制可通过函数 plot(x,y)实现，其中 $x=\tau, y=f(\tau)$（参见第 6.2 节 plot 函数的使用说明）。程序为：

```
n=1:150;                               %(1×150)
tau=linspace(-.5,.5,100);              %(1×100)
sn=sin(pi*n/4)./(pi*n/4);              %(1×150)
cntau=cos(2*pi*n'*tau);                %(150×100)
f=0.25*(1+2*sn*cntau);                 %(1×100)
plot(tau,f)
```

可以看出 f 是由向量相乘及点除运算得出的。点除运算得出(1×150)阶向量 ***sn***，而向量相乘得出式(2.11)所示的(150×100)阶矩阵 ***cntau***。依据式(2.13)，***sn*** * ***cntau*** 为一个行向量，其元素为对应每个 τ 值傅里叶级数的和。程序执行结果见图 2.5。

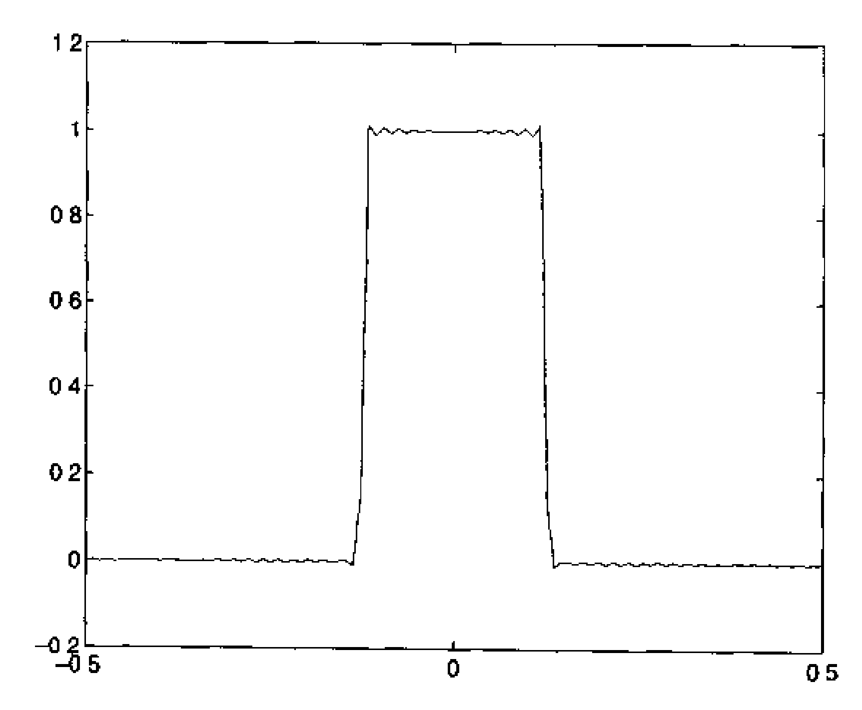


图 2.5 周期性脉冲的傅里叶级数求和

---

① 参见 H. P Hsu, *Applied Fourier Analysis*, Harcourt Brace Jovanovich, San Diego, CA, 1984 年。

例 2.4 正态累积分布函数

一个近似正态(高斯)累积概率分布函数,在 $0 \leqslant X \leqslant x$ 区间内,概率 P 由下式给出:[①]

$$P(x) = P(X \leqslant x) \approx 1 - \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \sum_{m=1}^{5} b_m (1 + 0.2316419x)^{-m} \quad 0 \leqslant x \leqslant \infty$$

其中 $0.5 \leqslant P(x) \leqslant 1$,并且:

$b_1 = 0.319\,381\,530$

$b_2 = -0.356\,563\,782$

$b_3 = 1.781\,477\,937$

$b_4 = -1.821\,255\,978$

$b_5 = 1.330\,274\,429$

由 $1 - P(|x|)$ 获得 $-\infty \leqslant x \leqslant 0$ 区域,其中 $0 \leqslant 1 - P(|x|) \leqslant 0.5$。

计算并绘制在 $-3 \leqslant x \leqslant 3$ 区间内增量 $\Delta x = 0.2$ 的累积分布的程序为:

```
b=[0.319381530 -0.356563782 1.781477937 -1.821255978 1.330274429];%(1×5)
m=1:length(b);                                  %(1×5)
x=0:0.2:3;                                      %(1×16)
mm=meshgrid(m,x);                               %(16×5)
bb=meshgrid(b,x);                               %(16×5)
temp=(1./(1+0.231641*x))';                      %(16×1)
zxx=meshgrid(temp,m)';                          %(16×5)
bzm=sum(bb.*(zxx.^mm))');                       %sum[(16×5)']→sum[(5×16)]→(1×16)
px=1-bzm.*exp(-0.5*x.^2)/sqrt(pi*2);            %(1×16)
plot(x,px,'k',-fliplr(x),fliplr(1-px),'k')
```

x 的长度为 16,m 的长度为 5。因此,由 meshgrid 函数创建的 3 个矩阵 ***mm***、***bb*** 和 ***zxx*** 的阶数均为(16×5)。如式(2.5)所示,(***bb***.*(***zxx***.^***mm***))′是一个(5×16)阶矩阵,而 sum((***bb***.*(***zxx***.^***mm***))′)是一个(1×16)阶向量。此外,exp(-0.5*x.^2)也是一个(1×16)阶向量。因此,***bzm***.*exp(-0.5*x.^2)即是一个(1×16)阶向量。由 1 减去该向量中的每一个元素,则每个结果都代表了对应 x 的 $P(x)$ 值,且 x 有 16 个值。plot 函数(详见第 6.2 节)使用两个三元组作为它的参数。每个三元组中的第 3 个量通知 plot 函数对线的每个部分都采用相同的颜色(本例中为黑色)。每个三元组中的前两个值给出了要显示的 x 和 y 坐标值。表达式 -fliplr(x)与创建新向量 $z = -3{:}0.2{:}0$ 相同,而表达式 fliplr(1 - *px*)使向量 1 - *px* 转置并创建了一个新向量,新向量的元素值与 -fliplr(x)中 x 取负值时相等。程序执行结果见图 2.6。

例 2.5 级数的双重求和

考虑下述双重求和形式:

$$w(x_i, y_j, t) = \sum_{n}^{N} \sum_{m}^{M} f_n(x_i) g_m(y_j) h_{nm}(t) \quad i = 1,2,\cdots s \quad j = 1,2,\cdots p$$

① 参见 M Abramowitz 和 I A Stegun, *Handbook of Mathematical Functions*, National Bureau of Standards, Applied Mathematics Series 55, U.S. Government Printing Office, Washington D.C., 1964 年,第 932 页。

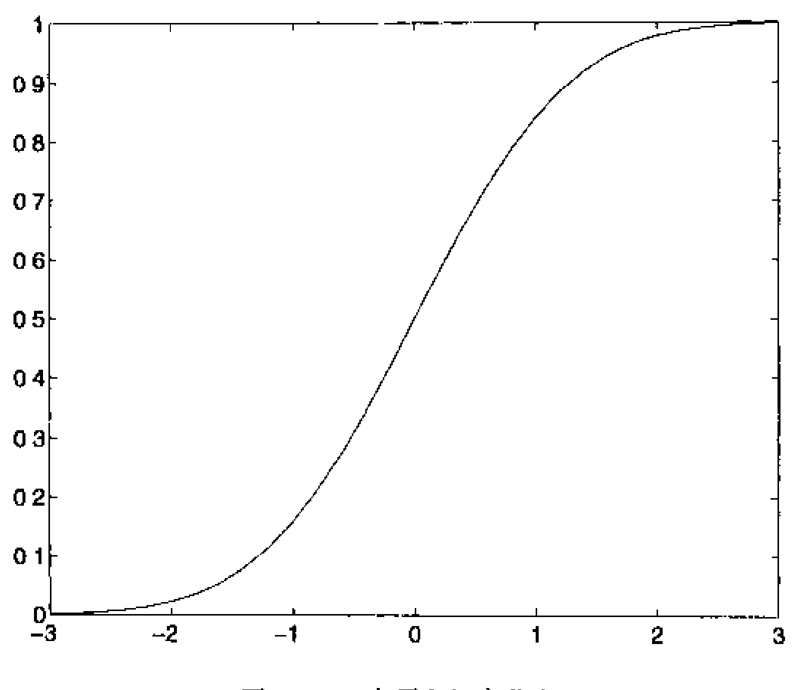


图 2.6　正态累积概率分布

其中 $\boldsymbol{x}=[x_1\ x_2\cdots\ x_s]$，$\boldsymbol{y}=[y_1\ y_2\cdots\ y_p]$。假设 $N=M=k$ 并令 $t=t_0$，则可按如下方式求取级数值。采用式(2.7)给出的 $(m\times n)$ 阶矩阵与元素为 d_{ij} 的 $(n\times p)$ 阶矩阵 $\boldsymbol{d}$ 相乘，则结果为 $(m\times p)$ 阶矩阵，如下所示：

$$\boldsymbol{w}=\boldsymbol{cd}=\boldsymbol{abd}=\begin{bmatrix}\sum_{l=1}^{n}\sum_{j=1}^{k}a_{1j}b_{jl}d_{l1} & \sum_{l=1}^{n}\sum_{j=1}^{k}a_{1j}b_{jl}d_{l2} & \cdots & \sum_{l=1}^{n}\sum_{j=1}^{k}a_{1j}b_{jl}d_{lp}\\ \sum_{l=1}^{n}\sum_{j=1}^{k}a_{2j}b_{jl}d_{l1} & \sum_{l=1}^{n}\sum_{j=1}^{k}a_{2j}b_{jl}d_{l2} & & \vdots\\ \vdots & & \ddots & \\ \sum_{l=1}^{n}\sum_{j=1}^{k}a_{mj}b_{jl}d_{l1} & \cdots & & \sum_{l=1}^{n}\sum_{j=1}^{k}a_{mj}b_{jl}d_{lp}\end{bmatrix}\rightarrow(m\times p) \tag{2.15}$$

用 $f_n(x)$ 构成 $(k\times s)$ 阶矩阵：

$$\boldsymbol{f}=\begin{bmatrix}f_1(x_1) & f_2(x_1) & \cdots & f_s(x_1)\\ f_1(x_2) & f_2(x_2) & \vdots & \\ \vdots & & \ddots & \\ f_1(x_k) & \cdots & & f_s(x_k)\end{bmatrix}\rightarrow(k\times s)$$

用 $g_m(y)$ 构成 $(k\times p)$ 阶矩阵：

$$\boldsymbol{g}=\begin{bmatrix}g_1(y_1) & g_1(y_2) & \cdots & g_1(y_p)\\ g_2(y_1) & g_2(y_2) & & \\ \vdots & & \ddots & \\ g_k(y_1) & \cdots & & g_k(y_p)\end{bmatrix}\rightarrow(k\times p)$$

用 $h_{nm}(t_0)$ 构成 $(k\times k)$ 阶矩阵：

$$
\boldsymbol{h} = \begin{bmatrix} h_{11}(t_0) & h_{12}(t_0) & \cdots & h_{1k}(t_0) \\ h_{21}(t_0) & h_{22}(t_0) & & \vdots \\ \vdots & & \ddots & \\ h_{k1}(t_0) & \cdots & & h_{kk}(t_0) \end{bmatrix} \rightarrow (k \times k)
$$

如果式(2.15)中的 $n = k$,则 $f'hg$ 的乘积即为$(s \times p)$阶矩阵 $\boldsymbol{w}$:

$$
\boldsymbol{w} = \begin{bmatrix} w(x_1, y_1, t_0) & w(x_1, y_2, t_0) & \cdots & w(x_1, y_p, t_0) \\ w(x_2, y_1, t_0) & w(x_2, y_2, t_0) & & \vdots \\ & & \ddots & \\ w(x_s, y_1, t_0) & \cdots & & w(x_s, y_p, t_0) \end{bmatrix} \rightarrow (s \times p)
$$

其中,

$$
w(x_i, y_j, t_0) = \sum_{n=1}^{k} \sum_{m=1}^{k} f_n(x_i) h_{nm}(t_0) g_m(y_j)
$$

为说明这一过程,考虑下列级数[①]。该级数描述了一个正方形薄膜相对于其原始位置 $w(\eta, \xi, 0) = \eta\xi(\eta - 1)(\xi - 1)$进行规则位移的无量纲响应。

$$
w(\eta, \xi, \tau) = \sum_{n=0}^{k} \sum_{m=0}^{k} \frac{\sin((2n+1)\pi\eta)\sin((2m+1)\pi\xi)}{(2n+1)^3 + (2m+1)^3} \cos\left(\pi\tau\sqrt{(2n+1)^2 + (2m+1)^2}\right)
$$

其中 $\eta = x/b$, $\xi = y/b$, b 为薄膜的边长, $\tau = ct_0/b$, c 为薄膜的波动速度, t 为时间。在 $\tau = 0$, $k = 10$时计算级数的程序为:

```
tau = 0;n = 0:10;m = n;xi = 0:0.1:1;eta = xi;
nden = meshgrid(2 * n + 1).^3,xi);
mden = meshgrid((2 * m + 1).^3,eta);
snx = sin(pi * (2 * n + 1)' * xi);
sme = sin(pi * (2 * m + 1)' * eta);
xterm = snx./nden';
yterm = sme./mden';
[n2,m2] = meshgrid(2 * n + 1,2 * m + 1,1);
w = xterm' * cos(pi * tau * sqrt(n2.^2 + m2.^2)) * yterm;
mesh(xi,eta,w)
```

向量 ***n***、***m***、***xi*** 和 ***eta*** 均为(1×11)阶向量。前两个 *meshgrid* 函数创建了两个(11×11)阶矩阵 ***ndem*** 和 ***mden***。向量相乘生成的 ***snx*** 和 ***sme*** 与矩阵 ***nden*** 和 ***mden*** 的点除运算得到 *xterm* 和 *yterm*。第 3 个 meshgrid 函数生成了两个(11×11)阶矩阵 ***n*2** 和 ***m*2**。***w*** 是一个(11×11)阶矩阵。mesh 函数与 surf 函数类似,最大的不同之处在于其形成的图形中块状区域为白色,见第 7.2 节。程序执行后所形成的表面见图 2.7。

---

① 参见 H P. Hsu,出处同上,第202页。

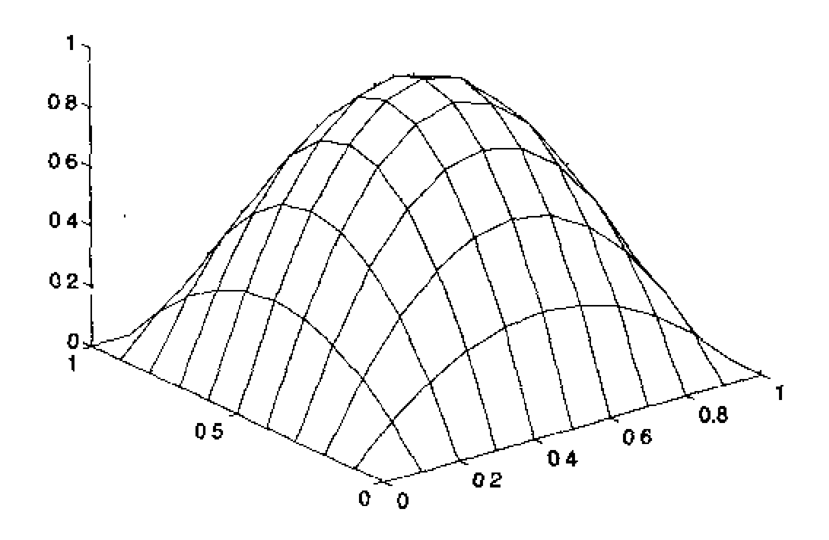


图 2.7　$t=0$ 时方形薄膜的形状

2.6.3 矩阵的逆

方阵 $\boldsymbol{a}$ 的逆运算：

$$\boldsymbol{a}^{-1}\boldsymbol{a}=\boldsymbol{a}\boldsymbol{a}^{-1}=\boldsymbol{I}$$

上式表明 $\boldsymbol{a}$ 为非奇异阵，也就是说，它的行列式（见第 2.6.4 节）不等于 0（$|a|\neq 0$）。上标"-1"代表逆。获得矩阵 $\boldsymbol{a}$ 的逆的表达式为：

```
inv(a)
```

或

```
a^-1
```

应注意 $1/a\neq a$^(-1)，$1/a$ 将产生错误。逆也可由第 2.6.5 节所述的反斜线运算符求得。

2.6.4 行列式

n 阶行列式定义如下：

$$|\boldsymbol{a}|=\begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & \vdots \\ \vdots & & \cdots & \\ a_{n1} & & \cdots & a_{nn} \end{vmatrix}$$

$n=2$ 时，

$$|\boldsymbol{a}|=a_{11}a_{22}-a_{12}a_{21}$$

$n=3$ 时，

$$|\boldsymbol{a}|=a_{11}a_{22}a_{33}+a_{12}a_{23}a_{31}+a_{13}a_{21}a_{32}-a_{13}a_{22}a_{31}-a_{11}a_{23}a_{32}-a_{12}a_{21}a_{33}$$

行列式的 MATLAB 表达式为：

```
det(a)
```

例如，若 $\boldsymbol{a}$ 定义为：

$$\boldsymbol{a}=\begin{bmatrix}1 & 3\\4 & 2\end{bmatrix}$$

则：

```
a=[1 3; 4 2]
d=det(a)
```

执行结果为 $\boldsymbol{d}=-10$。

一些工程应用中经常有如下这类问题：

$$|\boldsymbol{a}-\lambda\boldsymbol{b}|=0$$

其中 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 是$(n\times n)$阶矩阵，$\lambda_j(j=1,2,\cdots,n)$为方程的根(或特征值)。线性方程组的解可由下式获得：

```
lambda=eig(a,b)
```

例 2.6　多项式变换

有如下形式的多项式：

$$ax^2+by^2+cz^2+2dxy+2exz+2gyz$$

其中 a,b,c,d,e,g 为实数。多项式可变换为实对角阵形式：

$$r_1x'^2+r_2y'^2+r_3z'^2$$

其中 x',y',z'为另一坐标系，坐标系原点也在$(0,0,0)$，$r_1\geqslant r_2\geqslant r_3$ 为行列式：

$$|\boldsymbol{A}-r\boldsymbol{I}|=0$$

的根，$\boldsymbol{A}$ 为实对称阵：

$$\boldsymbol{A}=\begin{bmatrix}a & d & e\\d & b & g\\e & g & c\end{bmatrix}$$

考虑多项式：

$$4x^2+3y^2-z^2-12xy+4exz-8gyz$$

则：

$$\boldsymbol{A}=\begin{bmatrix}4 & -6 & 2\\-6 & 3 & -4\\2 & -4 & -1\end{bmatrix}$$

为求取根 r_j，可输入：

```
r=eig([4 -6 2; -6 3 -4;2 -4 -1])
```

执行后得出列向量 $r=[-4.0000\ \ -1.0000\ \ 11.0000]'$。注意这些根不是按给定顺序排列的，如想按预期给定的顺序排列可利用 sort 函数。但 sort 函数仅按由最小负值到最大正值的升序排列，因此，可利用下列表达式之一得到降序排列的数值，这些表达式可适用于

正实数与负实数的任意组合。

形式 #1:

```
r= - sort( - r)
```

可给出 $r=[11.0000\ \ -1.0000\ \ -4.0000]'$。

形式 #2:

```
r= flipud(sort(r))
```

也可给出 $r=[11.0000\ \ -1.0000\ \ -4.0000]'$。如 r 为行向量,则可将 flipud 更换为 fliplr。因此,实对角形式为:

$$11x'^2-y'^2-4z'^2$$

上述语句可简化为:

```
r= - sort( - eig([4 -6 2; -6 3 -4;2 -4 -1]))
```

2.6.5 方程组求解

考虑下列 n 阶方程组,n 个未知数 x_k, $k=1,2,\cdots,n$:

$$\begin{aligned}
a_{11}x_1+a_{12}x_2+\cdots+a_{1n}x_n&=b_1\\
a_{21}x_1+a_{22}x_2+\cdots+a_{2n}x_n&=b_2\\
&\vdots\\
a_{n1}x_1+a_{n2}x_2+\cdots+a_{nn}x_n&=b_n
\end{aligned}$$

方程组的矩阵形式可写为:

$$\boldsymbol{ax}=\boldsymbol{b}$$

其中 $\boldsymbol{a}$ 为$(n\times n)$阶矩阵:

$$\boldsymbol{a}=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n}\\ a_{21} & a_{22} & & \vdots\\ \vdots & & \ddots & \\ a_{n1} & \cdots & & a_{nn}\end{bmatrix}\rightarrow(n\times n)$$

且 $\boldsymbol{x}$ 和 $\boldsymbol{b}$ 分别为$(n\times 1)$阶向量:

$$\boldsymbol{x}=\begin{bmatrix} x_1\\ x_2\\ \vdots\\ x_n\end{bmatrix}\rightarrow(n\times 1)\quad \boldsymbol{b}=\begin{bmatrix} b_1\\ b_2\\ \vdots\\ b_n\end{bmatrix}\rightarrow(n\times 1)$$

其方程的解可在矩阵方程两边同乘以 $\boldsymbol{a}^{-1}$求出,即:

$$\boldsymbol{a}^{-1}\boldsymbol{ax}=\boldsymbol{a}^{-1}\boldsymbol{b}$$

$$\boldsymbol{x}=\boldsymbol{a}^{-1}\boldsymbol{b}$$

因为 $a^{-1}a = I$ 且 $Ix = x$，故推荐采用如下表达式求解此方程组：[1]

```
x = a \ b
```

其中反斜线运算符代表矩阵的除法，在 MATLAB 中称之为矩阵的左除。与以下公式相比，左除具有更好的数值稳定性：

```
x = a^-1 * b
```

或

```
x = inv(a) * b
```

以下面的方程组为例：

$$8x_1 + x_2 + 6x_3 = 7.5$$
$$3x_1 + 5x_2 + 7x_3 = 4$$
$$4x_1 + 9x_2 + 2x_3 = 12$$

其矩阵形式为：

$$\begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7.5 \\ 4 \\ 12 \end{bmatrix}$$

通过如下程序段求解方程组：

```
a=[8 1 6;3 5 7;4 9 2];
b=[7.5 4 12]';
x=a \ b
```

得 $x = [1.2931\ 0.8972\ \ -0.6236]'$。程序可简化为：

```
x=[8 1 6;3 5 7;4 9 2] \ [7.5 4 12]'
```

用下式检验：

```
z=a*x
```

得 $z = b$。

例 2.7　相关方程组的解

一正方形平板，四边固定，表面载荷均匀。则求解其静态变形时，首先应从下述无限方程组中通过舍去一些位获得常数 E_m 值：[2]

$$a_i E_i + \sum_{m=1,3,} b_{im} E_m = c_i \quad i = 1,3,5,\cdots$$

其中，

---

① 表示形式 $a \setminus b$ 也适用于 a 不是方阵时的情况，而 inv(a) 却仅适用于 a 为方阵的情况。也就是说，如果 a 是一个 $(m \times n)$ 阶矩阵，x 是一个 $(n \times 1)$ 阶向量，b 是一个 $(m \times 1)$ 阶向量，那么 $ax = b$，左除 $a \setminus b$ 得出 $x = cb$，其中 $c = (a'a)^{-1}a'$ 为矩阵 a 的准逆形式。

② 参见 S. Timoshenko 和 S. Woinowsky-Krieger, *Theory of Plates and Shells*, McGraw-Hill, New York, 1959 年，第 197～202 页。

$$a_\iota = \frac{1}{\iota}\left(\tanh\alpha_\iota + \frac{\alpha_\iota}{\cosh^2\alpha_\iota}\right)$$

$$b_{\iota m} = 8\iota\left(\pi m^3\left(1 + \frac{\iota^2}{m^2}\right)^2\right)^{-1}$$

$$c_\iota = \frac{4}{\pi^3\iota^4}\left(\frac{\alpha_\iota}{\cosh^2\alpha_\iota} - \tanh\alpha_\iota\right)$$

且 $\alpha_\iota = \iota\pi/2$。如果仅取方程组的前 4 个,则:

$$\begin{bmatrix} a_1 + b_{11} & b_{13} & b_{15} & b_{17} \\ b_{31} & a_3 + b_{33} & b_{35} & b_{37} \\ b_{51} & b_{53} & a_5 + b_{55} & b_{57} \\ b_{71} & b_{73} & b_{75} & a_7 + b_{77} \end{bmatrix}\begin{bmatrix} E_1 \\ E_3 \\ E_5 \\ E_7 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_3 \\ c_5 \\ c_7 \end{bmatrix}$$

可用下列程序求这 4 个方程的解:

```
m = 1:2:7;i = m;alp = m * pi/2;
ai = (tanh(alp) + alp ./(cosh(alp).^2)) ./i;                    %(1×4)
ci = 4. * (alp ./(cosh(alp).^2) - tanh(alp))./((pi^3) * i.^4);  %(1×4)
[ii,mm] = meshgrid(i,m);                                         %(4×4)
bim = (8/pi) * ii./((((1 + (ii.^2)./(mm.^2)).^2). * mm.^3);      %(4×4)
format long e
ee = (diag(ai) + bim) \ ci'
format short
```

meshgrid 函数得出两个如式(2.3)所示的(4×4)阶矩阵,通过这两个矩阵确定 *bim*,见式(2.4)。即使其下标为奇数,每个向量的长度都为 4。程序执行结果生成列向量,为简洁起见,对下列数值进行了简化处理。

ee→[-0.048000 0.004903 0.002296 0.001111]′

则 E_1 = ee(1,1) = -0.048 000;E_3 = ee(2,1) = 0.004 903;E_5 = ee(3,1) = 0.002 296;E_7 = ee(4,1) = 0.001 111。

练习

2.1 建立两个向量 ***a*** 和 ***b***,向量 ***a*** 元素为 $2n-1$,向量 ***b*** 元素为 $2n+1$,$n = 0,1,\cdots,7$。

(a) 求 ***a*** 与 ***b*** 的和?

(b) ***a*** 和 ***b*** 的区别是什么?

(c) 求 ***a′b*** 的乘积和它的行列式值?

(d) 求 ***ab′*** 的乘积?

2.2 给出向量 ***x*** = [17, -3, -47,5,29, -37,51, -7,19]。编程使 ***x*** 按照向量 ***y*** = [-3, -7, -37, -47,51,29,19,17,5]的形式重新排列。程序应适用于任意长度的向量。对所有向量,数值 0 与负数放在一起。也就是说,如果 0 为向量的一个元素,则它将作为 ***y*** 向量的第 1 个元素。

2.3 令 z = magic(5)。

(a) 按如下给定的顺序执行对 z 的运算。

(i) 第 2 列除以$\sqrt{3}$;

(ii) 把第 3 行元素加到第 5 行(第 3 行保持不变);

(iii) 把第 1 列与第 4 列相乘的结果放在第 1 列;

(iv) 令对角元素为 2。

(b) 如果(a)中求出的结果记为 q,则显示 qq'的对角元素值。

[答案:[486,104 189,7 300,44 522,111 024]'。]

(c) 显示 q 中每一元素的平方。

2.4 如图 2.8 所示曲柄滑块机构,滑块的位移为:

$$s = a\cos(\varphi) + \sqrt{b^2 - (a\sin(\varphi) - e)^2}$$

位移 s 是角度 φ(以角度表示)的函数,当 $a = 1, b = 1.5, e = 0.3, 0 \leqslant \varphi \leqslant 360°$时,使用 plot($\varphi$, s)函数绘制 s 的图形。

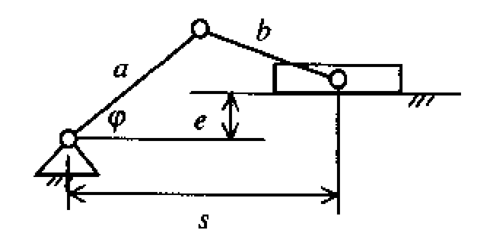


图 2.8 曲柄滑块机构

2.5 在矩形脉冲的周期级数中,将总功率的百分数 P 作为 N_H 的函数,其级数展开式为:

$$P = 100P_o/P_T\%$$

其中 P_T 为信号中的总功率:

$$P_o = 1 + 2\sum_{n=1}^{N_H} \frac{\sin^2(n\pi\tau_o/T)}{(n\pi\tau_o/T)^2}$$

τ_0/T 为脉冲宽度相对于周期的比值。若令 $\tau_0/T = 1/\sqrt{19}$,则 $P_T \approx 4.358\,9$。在 $2 \leqslant N_H \leqslant 25$ 的区间内绘制 P 的图形。

2.6 考虑下列乘积

$$S_N = \prod_{n=1}^{N}\left(1 - \frac{x^2}{n^2 - a^2}\right)$$

当 $N \to \infty$时,

$$S_\infty = \frac{a}{\sin\pi a\sqrt{a^2 + x^2}}\sin\left(\pi\sqrt{a^2 + x^2}\right)$$

相对误差定义为:

$$e_N = 100\,\frac{S_N - S_\infty}{S_\infty}\%$$

如果 x 的值在 1 到 5 之间以 0.5 的增量变化,且 $a = \sqrt{2.8}$,则当 $N = 100$ 时在 x 的 9 个取值处相对误差为多少?使用 prod 函数计算。

[答案：e_{10} = [1.000 1,2.264 3,4.061 0,6.417 6,9.370 7,12.967 0,17.264 2,22.333 0,28.258 8]。]

2.7 在威布尔概率密度函数[见第 14.2.2 节和练习 5.2(1)]中，参数 δ 的估计值可由下式获得：

$$\delta = \left[\frac{1}{n}\sum_{i=1}^{n} x_i^{\beta}\right]^{1/\beta}$$

其中 x_i 为 n 范围内的某一值，β 为另一已知参数。如果 x = [72 82 97 103 113 117 126 127 127 139 154 159 199 207]，且 $\beta = 3.644$，求取 δ 值。

[答案：$\delta = 144.274\ 1$。]

2.8 从球面坐标到笛卡儿坐标的转换公式为：

$$x = b\sin\phi\cos\theta$$

$$y = b\sin\phi\sin\theta$$

$$z = b\cos\phi$$

ϕ 在 $0 \leqslant \phi \leqslant 90°$ 的范围之内 10 等分，θ 在 $0 \leqslant \theta \leqslant 360°$ 的范围之内 24 等分。当 $b = 2$ 时，用 mesh(x, y, z)函数绘出这个半球形。

2.9 在 $0.1 \leqslant x \leqslant 1$ 的范围内将 x 值 5 等分，且当 $N = 25$ 时估算下列级数值：

$$\sum_{N\to-\infty}^{N\to\infty} \frac{1}{n^4 + x^4} = \frac{2\pi^4}{y^3}\frac{\sinh y + \sin y}{\cosh y - \cos y} \quad y = \pi x\sqrt{2}$$

并将这些值与实际值进行比较。

2.10 一类 n 阶贝塞尔函数由下式给出：

$$J_n(x) = \sum_{k=0}^{K\to\infty} \frac{(-1)^k (x/2)^{2k+n}}{k!\Gamma(k+1+n)}$$

在 $1 \leqslant x \leqslant 6$ 的范围内将 x 值 6 等分，当 $n = 2$，$K = 25$ 时求 $J_n(x)$ 的向量。伽马函数 Γ 和阶乘都可由 gamma 函数获得。将求出值与用 MATLAB 内置的 besselj(n, x)函数求出值进行比较。

2.11 如果，

$$\boldsymbol{X}'\boldsymbol{X} = \boldsymbol{I}$$

则矩阵被称为正交矩阵，因此，$(\boldsymbol{X}'\boldsymbol{X})^{-1} = \boldsymbol{I}$。证明下列矩阵为正交阵，可参见第 4.3.1 节。

$$\boldsymbol{w} = \frac{1}{2}\begin{bmatrix} -1 & -1 \\ 1 & -1 \\ -1 & 1 \\ 1 & 1 \end{bmatrix} \quad \boldsymbol{q} = \frac{1}{2}\begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

2.12 如图 2.9 所示，考虑平面上具有三个自由度的连杆机构。点 O_3 相对于固定坐标系 O_0 的位置和方向为：

$$T_3 = A_1 A_2 A_3$$

其中，

$$A_j = \begin{bmatrix} \cos\theta_j & -\sin\theta_j & 0 & a_j\cos\theta_j \\ \sin\theta_j & \cos\theta_j & 0 & a_j\sin\theta_j \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad j = 1,2,3$$

且，

$$T_3=\begin{bmatrix} u_x & v_x & 0 & q_x \\ u_y & v_y & 0 & q_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

元素 q_x 和 q_y 为 O_3 点相对于坐标原点 O_0 的 (x,y) 坐标。如果 $\theta_j=30°, j=1,2,3$ 且 $a_1=1, a_2=2, a_3=3$，求 O_3 点相对于坐标原点 O_0 的位置，并求出 (x_3,y_3) 轴系的方向。

[答案：$q_x=1.8660, q_y=5.2321$，x_3 平行于 y_0，y_3 平行于 x_0，但它们的方向相反。]

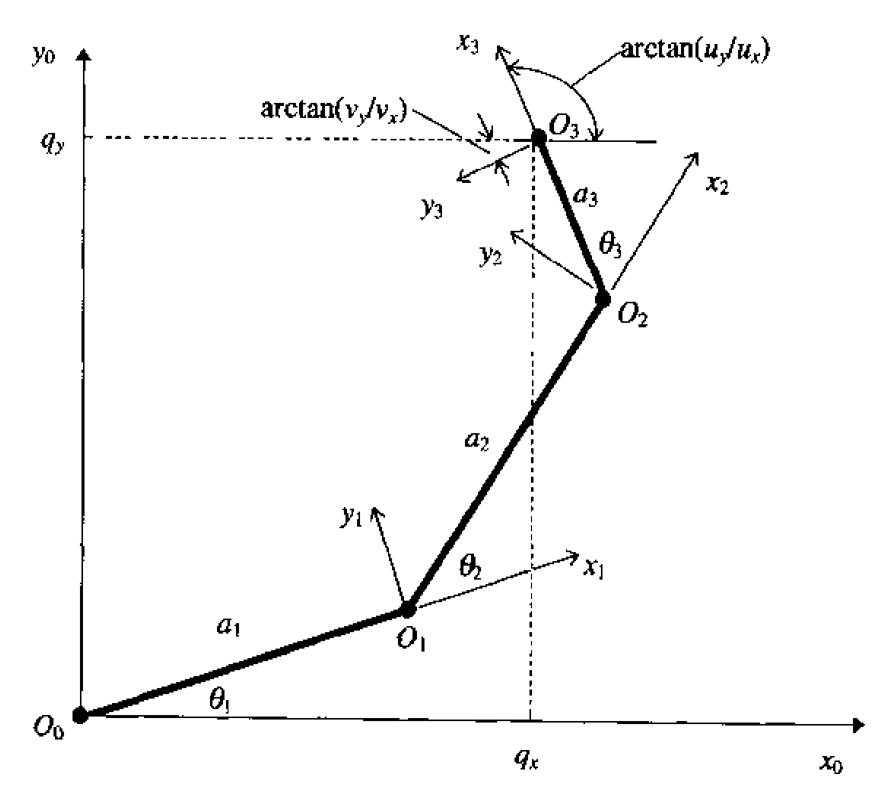


图 2.9　平面上具有三个自由度的连杆机构

2.13　在多重线性回归分析中，使用下列矩阵(见练习 14.13)：

$$H=X(X'X)^{-1}X'$$

如果，

$$X=\begin{bmatrix} 17 & 31 & 5 \\ 6 & 5 & 4 \\ 19 & 28 & 9 \\ 12 & 11 & 10 \end{bmatrix}$$

求对角阵 H。

[答案：对角阵 $H=[0.7294, 0.9041, 0.4477, 0.9188]'$。]

2.14　在给定的 τ 值范围内绘出下述级数[①] 的图形。除非特别指明，所有级数仅求出

① 参见 H P Hsu，出处同上。

200 项的和,求和运算不允许使用 sum 函数。

(a) 方波:

$$f(\tau)=\frac{4}{\pi}\sum_{n=1,3,5}\frac{1}{n}\sin(2n\pi\tau)\quad -\frac{1}{2}\leqslant\tau\leqslant\frac{1}{2}$$

(b) 锯齿波:

$$f(\tau)=\frac{1}{2}+\frac{1}{\pi}\sum_{n=1}\frac{1}{n}\sin(2n\pi\tau)\quad -1\leqslant\tau\leqslant 1$$

(c) 锯齿波:

$$f(\tau)=\frac{1}{2}-\frac{1}{\pi}\sum_{n=1}\frac{1}{n}\sin(2n\pi\tau)\quad -1\leqslant\tau\leqslant 1$$

(d) 三角波:

$$f(\tau)=\frac{\pi}{2}-\frac{4}{\pi}\sum_{n=1}\frac{1}{(2n-1)^2}\cos((2n-1)\pi\tau)\quad -1\leqslant\tau\leqslant 1$$

(e) 正弦整流波:

$$f(\tau)=\frac{2}{\pi}+\frac{4}{\pi}\sum_{n=1}\frac{1}{1-4n^2}\cos(2n\pi\tau)\quad -1\leqslant\tau\leqslant 1$$

(f) 半正弦波:

$$f(\tau)=\frac{1}{\pi}+\frac{1}{2}\sin\pi\tau-\frac{2}{\pi}\sum_{n=2,4,6,}^{106}\frac{\cos n\pi\tau}{n^2-1}\quad -2\leqslant\tau\leqslant 2$$

(g) 指数:

$$f(\tau)=\frac{e^{2\pi}-1}{\pi}\left[\frac{1}{2}+\sum_{n=1}^{250}\frac{1}{1+n^2}(\cos n\tau-n\sin n\tau)\right]\quad 0\leqslant\tau\leqslant 4\pi$$

使用 350 个 τ 值显示结果。

(h) 梯形波:

$$f(\tau)=\frac{4}{a^2}\sum_{n=1,3,5,}^{105}\frac{\sin n\pi a}{(\pi n)^2}\sin n\pi\tau\quad -2\leqslant\tau\leqslant 2$$

令 $a=0.25$。

2.15 考虑两个级数:[①]

$$S_{1N}=\sum_{n=1}^{N}\frac{\cos(n\theta)}{n^2+a^2}\qquad 0<\theta<\pi$$

$$S_{2N}=\sum_{n=1}^{N}\frac{n\sin(n\theta)}{n^2+a^2}\qquad 0<\theta<2\pi$$

其中,当 $N\to\infty$ 时:

$$S_{1\infty}=\frac{\pi\cosh[a(\pi-\theta)]}{2a\sinh\pi a}-\frac{1}{2a^2}\qquad 0<\theta<\pi$$

$$S_{2\infty}=\frac{\pi\sinh[a(\pi-\theta)]}{2\sinh\pi a}\qquad 0<\theta<2\pi$$

相对误差定义为:

$$e_{jN}=100\frac{S_{jN}-S_{j\infty}}{S_{j\infty}}\%\qquad j=1,2$$

① 参见 L. B. Jolley, 出处同上。

如果 θ 在 10°到 80°的范围内以 10°的增量变化，且 $\alpha=\sqrt{3}$，则当 $N=25$ 时，求这两个级数对应 8 个 θ 值的相对误差（不使用 sum 函数）。

[答案：$e_1=[-1.2435, 0.8565, 0.8728, -1.9417, -0.9579, -8.1206, 0.7239, 1.1661]$，$e_2=[8.0538, 10.4192, -8.9135, -5.4994, 12.9734, -0.5090, -17.2259, 11.2961]$。]

2.16 边缘 $\eta=1$ 处为恒温的矩形平板，其无量纲稳态温度分布函数为：[1]

$$T(x,y)=\frac{4}{\pi}\sum_{n=1,3,5}^{\infty}\frac{\sinh(n\pi\alpha\eta)}{n\sinh(n\pi\alpha)}\sin(n\pi\xi)$$

其中 $\eta=x/d$，$\xi=y/b$，d 和 b 分别为平板在 x 和 y 方向的长度，$\alpha=d/b$，$0\leqslant\eta\leqslant1$ 且 $0\leqslant\xi\leqslant1$。令 $\Delta\eta=\Delta\xi=1/14$，当 $\alpha=2$ 时，利用 mesh 函数显示整个平板的温度分布。不使用 sum 函数来获得结果。

2.17 在一个初始速度为 0 的弦中，初始位移如下：

$$u(\eta,0)=\frac{\eta}{a}\qquad 0\leqslant\eta\leqslant a$$

$$u(\eta,0)=\frac{1-\eta}{1-a}\quad a\leqslant\eta\leqslant1$$

波的直线传播位移公式由下式给出：

$$u(\eta,\tau)=\frac{2}{a\pi(1-a)}\sum_{n=1}^{N\to\infty}\frac{\sin n\pi a}{n^2}\sin(n\pi\eta)\cos(n\pi\tau)$$

当 $N=50$，$a=0.25$，$\Delta\eta=0.05$，$\Delta\tau=0.05$，$0\leqslant\tau\leqslant2$ 时显示 $u(\eta,\tau)$，不使用 sum 函数。结果如图 2.10 所示，它是通过使用图形窗口中的“旋转”图标得到的。

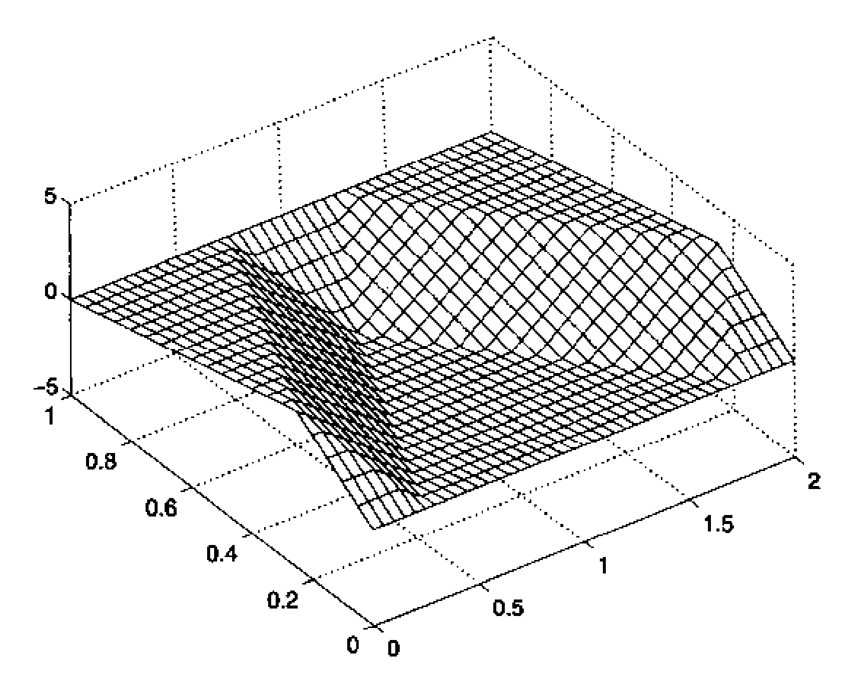


图 2.10 在一根弦中初始位移的传播

2.18 给出如下方程组：

$$16s+32u+33p+13w=91$$

---

[1] 参见 H. P. Hsu，出处同上。

$$5s + 11u + 10p + 8w = 16$$
$$9s + 7u + 6p + 12w = 5$$
$$34s + 14u + 15p + w = 43$$

求 s, u, p, w 的值,并求出系数的行列式。

[答案:$s = -0.125\,8, u = -8.713\,3, p = 11.287\,5, w = -0.050\,0$。行列式 = 7 680。]

2.19 考虑两个由不同物质构成的长圆筒,其中一个圆筒正好套在另一个圆筒里面。里侧圆筒的内径为 a,外径为 b;外侧圆筒的内径为 b,外径为 c。里侧圆筒的弹性模量和泊松比分别为 E_1 和 ν_1,外侧圆筒的弹性模量和泊松比分别为 E_2 和 ν_2。径向应力 σ_{ri},环向应力 $\sigma_{\theta\theta}$ 和径向位移 u_r 分别由下式给出:

$$\sigma_{ri}(r) = \frac{A_i}{r^2} + B_i$$

$$\sigma_{\theta\theta i}(r) = \frac{-A_i}{r^2} + B_i \tag{a}$$

$$u_{ri}(r) = \frac{-(1+v_i)}{rE_i}A_i + \frac{(1-v_i)}{E_i}rB_i$$

其中 $i = 1$ 表示里侧圆筒,$i = 2$ 表示外侧圆筒。

如果外侧圆筒的外表面有一个压缩的径向位移 U_o,而里侧圆筒的内表面没有径向应力,则下列 4 个边界条件可用于确定 A_i 和 B_i,$i = 1,2$:

$$\begin{aligned} &\sigma_{r1}(a) = 0 \\ &\sigma_{r1}(b) = \sigma_{r2}(b) \\ &u_{r1}(b) = u_{r2}(b) \\ &u_{r2}(c) = -U_o \end{aligned} \tag{b}$$

将式(a)代入式(b)得到下列方程组:

$$\begin{bmatrix} 1 & a^2 & 0 & 0 \\ 1 & b^2 & -1 & -b^2 \\ -(1+v_1) & (1-v_1)b^2 & (1+v_2)E_1/E_2 & -(1-v_2)b^2E_1/E_2 \\ 0 & 0 & -(1+v_2) & (1-v_2)c^2 \end{bmatrix} \begin{Bmatrix} A_1 \\ B_1 \\ A_2 \\ B_2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ -U_oE_2c \end{Bmatrix}$$

当 $\nu_1 = \nu_2 = 0.4$, $E_1 = 3 \times 10^5$ psi, $E_2 = 3.5 \times 10^4$ psi, $U_o = 0.01$ in, $a = 0.192$ in, $b = 0.25$ in, $c = 0.312$ in 时,求里侧圆筒和外侧圆筒在 $r = b$ 处的环向应力。

[答案:$\sigma_{\theta\theta 1}(b) = -9\,571.8$ psi, $\sigma_{\theta\theta 2}(b) = -1\,989.3$ psi。]

第3章　数据的输入与输出

本章将介绍在MATLAB命令窗口中显示数值结果的方法,以及如何存储或从文件中查找数据的方法。

3.1　字符串(文字)和注释的输出

在MATLAB中允许矩阵的创建、存贮和操作,这些矩阵可以是字母、数字和一些特殊字符任意组合而成的字符串。字符串的定义是把字符串放在单引号内,除此以外,其他与向量和矩阵的定义类似。并且字符串的每个字符在向量或矩阵中占据一个元素的位置。

考虑下面的例子,令 *s* 等于字符串'testing123'。MATLAB将把这个字符串定义为向量:

```
s='testing123'
```

或

```
s=['testing123']
```

在向量 *s* 中,单引号内的每个字符占一个存储地址。因此写入:

```
s(7)→g
s(3:6)→stin
fliplr(s)→321gnitset
```

并且 length(*s*)=10。字符串的连接与数值相似。则:

```
sc=[s fliplr(s)]
```

产生字符串:

```
testing123321gnitset
```

而

```
scs=[s;fliplr(s)]
```

是如下的矩阵:

```
scs(1,:)→testing123
scs(2,:)→321gnitset
```

注意 scs 每行具有相同数目的字符(列数)。

如果将字符串放在矩阵中的每一行,那么会有一个很方便的方法得到字符串表达式。这要求每一行必须包含相同数目的字符。当字符串表达式的长度不相等时,可以使用空格填充字符串的其他部分,如:

```
lab=['first';'last';'middle']
```

那么,

```
lab(1,:)→first b
lab(2,:)→last bb
lab(3,:)→middle
```

其中 *b* 为空格。幸运的是,MATLAB 通过函数 str2mat 实现这种填充。因此,上面的表达式可以很方便地用下式代替:

```
lab = str2mat('first','last','middle')
```

其中每一个字符串表达式为矩阵 ***lab*** 的一行,形成一个(3×6)维数组。字符串末尾的空格通过使用函数

```
deblank
```

被删除。

将数值转换为字符串可以使用函数:

```
num2str
```

令 *num* 为数值、数值矩阵或是数值及矩阵的表达式。用下式可把它转换为字符串:

```
z = num2str(num)
```

其中 *z* 是一个字符串变量。

函数 num2str 经常用来在 MATLAB 命令窗口或图形中显示数值输出的注释。一个典型结构可将转换的数值与一些相关的文本关联起来。因此,若 *num* 是用千克表示的重量,则在 MATLAB 命令窗口中用 disp 函数显示 *num* 如下:

```
num = 12.567;
z = num2str(num);
disp(['Product weight = ' z ' kg'])
```

或者,更为紧凑的形式是:

```
num = 12.567;
disp(['Product weight = 'num2str(num)'kg'])
```

执行后,显示在 MATLAB 命令窗口中的结果为:

```
Product weight = 12.567 kg
```

实质上,这个字符串是长度为 26 的向量。注意空格没有被删除。

若 *num* 是重量的向量,那么,

```
num = [12.567 3.458 9 111];
disp(['Product weight = 'num2str(num)'kg'])
```

显示:

```
Product weight = 12.567 3.458 9.111 kg
```

然而,可以使用下面的程序建立与每个 *num* 值一致的注释:

```
num = [12 567 3 458 9.111];
```

```
n = length(num);
disp([repmat('Product weight = ',n,1)num2str(num') repmat('kg',n,1)])
```

执行后显示：

```
Product weight = 12.567 kg
Product weight = 3.458 kg
Product weight = 9.111 kg
```

若不需要显示 *num* 的注释，则，

```
num = [12.567   3.458   9.111];
disp(num)
```

在 MATLAB 命令窗口中显示：

```
12.5670   3.4580   9.1110
```

而，

```
num = [12.567 3.458 9.111];
disp(num')
```

显示：

```
12.5670
3.4580
9.1110
```

MATLAB 允许用户指定数值转换为字符串时的位数：

```
num2str(a,N)
```

其中 a 是要转换的数值，N 是数值的位数。若指定的位数小于小数点左边的位数，则 MATLAB 将数值转换为指数描述形式，且有效数字等于 N。

考虑下面的例子，$a = 1\,000\pi = 3\,141.592\,653\,589$，则：

```
num2str(a,1)→3e+003
num2str(a,3)→3.14e+003
num2str(a,4)→3142
num2str(a,5)→3141.5
num2str(a,8)→3141.5927
```

注意位数不包括小数点。

函数 fprintf 可以在 MATLAB 命令窗口中显示有格式的数据，它相对于函数 disp 的优势在于它能够控制数值的显示格式。在 MATLAB 命令窗口中，函数 fprintf 的语法结构为：

```
fprintf(1,'%….',variables)
```

其中第一个参数为'1'，表示在 MATLAB 命令窗口中输出，而引号内的参数是变量格式的附加说明。格式说明的顺序对应着变量的顺序。符号% 在每个具体格式说明之前。常用的格式说明形式为：

```
x.yf
```

f 为格式类型中的一种。fprintf 的其他格式见 MATLAB 的帮助文件。x 为给定数值整数部分的位数,而 y 是小数点后的位数。通过下面的向量可说明 fprintf 的几种不同用法。

```
num = [12  - 14 3 458 0 11167];
```

用 fprintf 函数在一行中显示该向量:

```
num = [12  - 14 3.458 0.11167];
fprintf(1,' %5 3f ',num)
```

结果为:

```
12.000  -14.000  3.458  0.112 >>
```

注意 *num*(1)和 *num*(2)的值加了 3 个 0,而 *num*(4)在小数点后则近似取了 3 位数。另外, >> 表明数据已经在一行中显示出来,可以执行其他的 MATLAB 表达式。若想得到下一行,按 Enter 键即可。如果要把这 4 个数值作为一列来显示,可以采用定义符"\ *n*",如下所示:

```
num = [12  - 14 3 458 0.11167];
fprintf(1,'%5.3f \ n',num)
```

结果为:

```
12.000
-14.000
3.458
0.112
```

若想得到与给定精度相同的 4 个数值,则:

```
num = [12  - 14 3.458 0.11167];
fprintf(1,'%2.0f %2.0f %5.3f %5.5f ',num)
```

其中每个 f 和% 之间都有两个空格,这样数值就被两个空格分开了。表达式执行后产生:

```
12  -14   3.458   0.11167 >>
```

可以用如下方法对每一个数值进行注释:

```
num = [12  - 14 3.458 0.11167];
fprintf(1,'weight = %2.0f kg pressure = %2.0f Pa time = %5.3f s length = %5.5f
          m \ n',num)
```

结果为:

```
weight = 12 kg pressure = - 14 Pa time = 3.458 s length = 0.11167 m
```

如想按列显示数值,则程序为:

```
num = [12  - 14 3.458 0.11167];
fprintf(1,'weight = %2.0f kg \ npressure = %2 0f Pa \ ntime = %5.3f s \ nlength
          = %5.5f m \ n',num)
```

结果为:

```
weight = 12 kg
pressure = -14 Pa
time = 3.458 s
length = 0.11167 m
```

若让每个数值按相同的格式显示,那么可以简化格式说明并产生注释输出,程序如下:

```
num = [12 -14 3.458 0.11167];
nn = 1:length(num);
fprintf(1,'x(%1.0f) = %7.5f\n',[nn;num])
```

结果为:

```
x(1) = 12.00000
x(2) = -14.00000
x(3) = 3.45800
x(4) = 0.11167
```

3.2 用 INPUT 函数输入数据

数组可以按程序或函数限定的格式由用户用 input 输入。此外,input 还可以在 MATLAB 命令窗口中显示信息以提示用户输入什么。然而,数据的具体形式依赖于数据是标量、向量还是矩阵,以及它们是数值还是字符串。举例说明以下这几种情况。其他数据输入方法见第 3.3 节和第 5.2 节。

3.2.1 标量的输入

输入一个数值时,可以采用:

```
InputData = input('Enter the temperature in degrees C:');
```

在 MATLAB 命令窗口中显示:

```
Enter the temperature in degrees C: 121.7
```

其中数值 121.7 由用户输入。表达式末尾的分号可禁止系统对于输入数据的回显。

在同一表达式中可以执行用户输入数据的计算。例如:

```
InputData = input('Enter the starting angle in degrees:') * pi/180;
```

结果显示:

```
Enter the starting angle in degrees:45
```

其中数值 45 是由用户输入的。然而, *InputData* 的数值为 0.785 4($=45\pi/180$)。

又如,考虑温度从摄氏度向华氏度的变换,输入:

```
InputData = 1.8 * input('Enter the temperature in degrees C:') + 32;
```

结果显示:

```
Enter the temperature in degrees C:100
```

其中数值 100 由用户输入。而 *InputData* 的数值为 212。

如希望将提示信息显示在几行中,则可以在单引号内的适当位置上加入定义符"\n",即:

```
InputData = input('Enter the starting angle \n in degrees:') * pi/180;
```

则结果显示：

```
Enter the starting angle
in degrees:
```

3.2.2 字符串的输入

输入一个字符串时，可以采用：

```
InputData = input('Enter file name,including its extension:','s');
```

在 MATLAB 命令窗口中显示：

```
Enter file name,including its extension:DataSet3.txt
```

其中字符串 *DataSet3.txt* 由用户输入。注意这里不需要加单引号，这是由 input 函数中第二项参数's'决定的。

3.2.3 向量的输入

输入由数值构成的向量时，可以采用：

```
InputData = input('Enter the temperatures in degrees C:');
```

在 MATLAB 命令窗口显示：

```
Enter the temperatures in degrees C:[120 141 169 201]
```

其中向量[120 141 169 201]由用户输入。这里的方括号是必需的。若要输入列向量，则用户应输入[120 141 169 201]'。

3.2.4 矩阵的输入

输入由数值构成的矩阵时，可以采用：

```
InputData = input('Enter the three temperatures in degrees C \nat levels 1 and 2:');
```

在 MATLAB 命令窗口显示：

```
Enter the three temperatures in degrees C
at levels 1 and 2:[67 35 91;44 51 103]
```

其中数组[67 35 91;44 51 103]由用户输入。变量 *InputData* 是一个(2×3)维数组。

3.3 数据文件的输入与输出

如前所述，数据输入的一种方法是通过程序用 input 函数完成。第二种方法是在程序中定义数据，就如同第 2.3 节和第 2.4 节中讨论的一样。如第 5 章所述，创建数据的语句也可以在函数中出现。实际上，可以定义只包含数据的函数。见第 5.2 节中的例子。

其他输入数据的方法是将数据存放在 ASCII 文本文件之中，然后调用 load 函数。load 函数按行读取数据，每行由回车符分隔且每个数值之间用空格或逗号分隔。每行中数据的列数必须

相同,而每列中的行数也必须相等。这些要求与矩阵的创建非常相似。这里的回车符代替了分号。在创建行向量时,输入数据可以不用回车符。而创建列向量时每一个数值后都要跟一个回车符。

下面举例说明 load 函数的两种使用方法。假设数据按下面的形式存放在 *DataSection33.txt* 文件中:

```
11 12 13
21 22 23
31 32 33
41 42 43
```

load 函数的用途在于将不带扩展名(后缀".txt")的主文件名变为变量名,将文件中数据变为向量或矩阵元素。因此,程序中名为 *DataSection33* 的变量是一个(4×3)阶的数值矩阵,它在程序中的使用就如同一个已放置在等号左边的名为 *DataSection33* 的变量一样。

函数语句可为:

```
load DataSection33.txt
```

或

```
load 'DataSection33.txt'
```

或

```
load('DataSection33.txt')
```

函数应用的前提是假设文件的路径已经指定,否则,可以使用 cd 命令或选择 File 下拉菜单的"Set Path"子菜单设置文件路径。这三种函数形式应用于创建程序时,文件名已知且不再更改的情况。因此,如希望对矩阵中的每一个元素取平方,可输入:

```
load 'DataSection33.txt'
y = DataSection33.^2
```

结果显示:

```
121    144    169
441    484    529
961    1024   1089
1681   1764   1849
```

另一方面,如在不同文件中运算数据,而且具有不同的文件名,则必须要使用不同的方法。因此进行数据运算时,用户首先应通过程序或函数输入文件名,如前所述;程序执行后,给定文件中的所有元素都将会取平方。相应程序为:

```
FileName1 = input('Enter file name containing data (including suffix):','s');
load(FileName1);
m = findstr(FileName1,'.');
data1 = eval(FileName1(1:m-1));
y = data1.^2
```

函数 findstr 用来查找单引号内字符串中第一个出现目标字符的位置,并返回它的值,上例中运算符为句号(.)。这里使用它是为了限制组成 *FileName1* 的字符串,但不包括句号,因此字符串长度为 $m-1$。这样就删除了文件名后缀。然而,去除后缀的 *FileName1* 对于程序表达式来

说仍然是未知的，因此还必须把它转化为数值。这项工作由 eval 函数来完成，它可计算参数引用中字符串的值。

如果要把程序中的数值保存到文件之中，可使用：

```
save
```

假如要把文件 *DataSection33.txt* 中每个值的平方存成 ASCII 文本文件，程序为：

```
load 'DataSection33.txt'
y = DataSection33.^2
save SavedDataSection33.txt' y - ascii
```

结果生成一个包含下列内容的文本文件：

```
1.2100000e+002 1.4400000e+002 1.6900000e+002
4.4100000e+002 4.8400000e+002 5.2900000e+002
9.6100000e+002 1.0240000e+003 1.0890000e+003
1.6810000e+003 1.7640000e+003 1.8490000e+003
```

当给定文件名时，MATLAB 把文件存放在默认的目录中。而要想把文件存放在指定的目录中，则必须要给出完整的路径名。例如：

```
load 'DataSection33.txt'
y = DataSection33.^2;
save 'c:\Matlab mfiles/Matlab results\SavedDataSection33.txt' y - ascii
```

注意，一定要用单引号将完整的路径名和文件名引起来，这是因为在目录名上出现了冒号和空格。另外，若想在这个文件中保存其他的值，可以在 save 语句中按如下方法附加相应的变量名。若让上面的程序还能计算 *DataSection33.txt* 中数值的平方根，则程序变为：

```
load 'DataSection33.txt'
y = DataSection33.^2;
z = sqrt(DataSection33);
save 'c:/Matlab mfiles/Matlab results/SavedDataSection33.txt'y z-ascii
```

文件中的数据是一个(8×3)阶矩阵。前 4 行由 y 确定，后 4 行由 z 确定。

第 6.1 节给出了图形的存储过程。

练习

3.1　由下式产生黄金分割数：

$$F_n=\frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n-\left(\frac{1-\sqrt{5}}{2}\right)^n\right]\quad n=0,1,2,\cdots$$

使用 fprintf 和 disp 产生前 16 个数值，并在 MATLAB 命令窗口中显示如下内容：

```
F0 = 0
F1 = 1
F2 = 1
F3 = 2
⋮
F15 = 610
```

第4章　程序流程控制

本章介绍控制程序表达式运算顺序的多种方法，以及一系列典型的用以实现条件和逻辑控制的算法。

4.1　概述

程序表达式的运算顺序有4种结构控制命令：while、if、for 和 switch。每当这些命令出现时，在程序中的稍后部分都必须要跟随一个 end 语句。只有当所有条件满足时，控制结构命令与 end 命令之间的语句才会被执行。每一种控制结构命令可以随意相互之间嵌套或自身嵌套，称为嵌套结构。

控制结构通常把关系运算或逻辑运算的结果作为判定条件。当条件满足时控制语句将程序转向特定部分执行相应的语句。MATLAB 的部分关系运算符和逻辑运算符见表4.1。

表4.1　部分关系运算符及逻辑运算符

条件	数学符号	MATLAB 符号
关系运算符		
等于	=	= =
不等于	≠	~ =
小于	<	<
大于	>	>
小于等于	≤	< =
大于等于	≥	> =
逻辑运算符		
与	AND	&
或	OR	|
非	NOT	~

采用控制结构时，建议控制结构命令与 end 命令对齐，中间部分缩进，形成一个锯齿状结构，这将增强程序或函数的可读性。采用嵌套结构时，内部嵌套部分也采用锯齿状结构，并保证整个嵌套结构都是锯齿状的。这样的语句结构在 MATLAB 的编辑/调试窗口可以自动实现。

利用表4.1中的逻辑运算符可产生这样的一个逻辑函数：当逻辑表达式结果为真时输出为1，结果为假时输出为0。要产生如下的函数 $g(x)$：

$$g(x) = f(x) \qquad a \leqslant x < b$$
$$= 0 \qquad \text{其他}$$

则逻辑表达式可写为：

```
y = (a< = x & x<b);
```

a 和 b 的值已知,且,

(a< =x & x<b)

$x \geq a$ 和 $x<b$,逻辑表达式的值为 1。当 x 为其他值时,逻辑表达式的值为 0。因此,如果令 $a=-1, b=2, f(x)=e^{x/2}$,且 $x=[-4\ \ -1\ 1\ 4]$,那么采用这一逻辑表达式的程序如下:

```
a= -1;b=2;
x=[-4 -1 1 4];
gofx = exp(x/2).*(a<=x & x<b)
```

最后计算得出 $gofx=[0, 0.6065, 1.6487, 0]$。

4.2 程序流程控制

4.2.1 while 循环

while 循环无限次执行一条或多条语句,当条件满足时停止循环。其语法结构为:

```
while 条件
  语句
end
```

其中,定义"条件"的表达式中可包含由"语句"计算得出的一个或多个变量。

现在给出两个 while 循环的例子。第一个例子用来确保用户输入数据在规定的区间内。第二个例子用于判定是否收敛。

例 4.1　正确输入数据

下列摘录的程序段要求用户连续输入 1 到 8 之间的数,若所输入的数值超出范围,则程序终止。其中 input 函数将引号内的部分显示于 MATLAB 窗口,并等待用户输入数据,再将所输入的数据赋给变量 *nfnum*。符号"|"代表逻辑或。

```
nfnum=0;
while(nfnum<1)|(nfnum>8)
  nfnum = input('Enter a number from 1 to 8: ');
end
```

对变量 *nfnum* 赋初值(本例中为 0),while 命令检查条件表达式"(*nfnum* < 1)|(*nfnum* > 8)"是否满足,并以此决定是否进入 while 循环。执行完 end 语句前的最后一行语句后,程序返回,重新判断 while 条件表达式是否满足。条件不满足,重复执行 while 后面的语句,否则执行 end 后面的语句。注意,在进入 while 循环块前一定要给变量 *nfnum* 赋初值。

例 4.2　级数收敛性判定

计算并显示级数 S_n 的项数 N, S_n 如下:

$$S_N=\sum_{n=1}^{N}\frac{1}{n^2}$$

当收敛到终值 $S_\infty=\pi^2/6$ 的 0.01% 以内时,项数 N 为多少。程序如下:

```
series = 1;k = 2;exact = pi^2/6;
while abs((series - exact)/exact) > = 1e - 4
  series = series + 1/k^2;
  k = k + 1;
end
disp(['# terms = 'num2str(k - 1)])
```

给变量 *series* 赋初值(本例中为 1),while 语句检查条件 abs((*series* - *exact*)/*exact*)是否满足,决定是否执行 while 循环。执行完 end 的前一条语句后,程序返回到 while 开始处检查 while 条件是否满足。如果条件不满足,接着执行 while 下面的语句;否则执行 end 下面的语句。本例在 MATLAB 命令窗口中显示结果。

当出现(*series* - *exact*)为负值的情况时,判断仍然有效。使用((*series* - *exact*)/*exact*)的绝对值检查条件,负值小于 10^{-4} 是显然成立的,而它的绝对值却不一定小于 10^{-4}。采用绝对值条件后不用再考虑 *series* 值是从正向还是从负向趋近极限值。在构建 while 循环的条件表达式时必须要小心,因为一旦表达式不严格,循环将会无限次地执行下去,直至按下 Ctrl^C 组合键才能终止。

4.2.2 if 语句

if 语句的常见形式如下:

```
if 条件 # 1
  表达式 # 1
elseif 条件 # 2
  表达式 # 2
else
  表达式 # 3
end
```

若条件 # 1 满足,则执行表达式 # 1,然后执行与之对应的 end 语句后面的语句。当条件 # 1 不满足时,检验条件 # 2 是否满足。如果此时条件 # 2 满足,则执行表达式 # 2,然后执行与之相应的 end 语句后面的语句。如果条件 # 1 和条件 # 2 都不满足,则执行表达式 # 3,然后执行 end 语句后面的语句。这里 elseif 和 else 语句是可选的,且程序中可同时出现多个 elseif 语句。

下段程序解释 if 语句结构的用法。变量 *j*、*x* 和 *nnum* 已在前面赋值或已在前面的程序中由计算过程赋值。

```
if j == 1
   z = sin(x);        ←— 仅当j=1时执行。
   if nnum<=4         ←— 这个if语句仅当j=1时才考虑。
     nr = 1;       ]
     nc = 1;       ]  ←— 这两条语句仅当j=1且nnum≤4时才被执行。
   else
     nr = 1;       ]
     nc = 2;       ]  ←— 这两条语句仅当j=1且nnum>4时才被执行。
   end
else
   nr = 2;         ]
   nc = 1,         ]  ←— 这两条语句仅当j≠1时才被执行。
end
```

注意，这里内嵌了一个 if 语句，因此需要书写另一个 end 语句。并且应注意如何在程序的不同嵌套结构内部缩进表达式以增强代码的可读性。

例 4.3　疲劳强度计算

根据给定的校正系数计算金属的疲劳强度。

因素	范围	校正系数
受力方式	弯曲	$C_{load}=1$
	轴向	$C_{load}=0.70$
尺寸	$d\leqslant 8$ mm	$C_{size}=1$
	$8\leqslant d\leqslant 250$ mm	$C_{size}=1.189d^{-0.097}$
温度	$T<450$℃	$C_{temp}=1$
	$450\leqslant T$℃	$C_{temp}=1-0.0032(T-840)$

判别这些因素的程序如下所示。其中变量 *lode*、*d* 和 *temp* 已在前面的程序中赋值或计算出结果。*lode* 为字符型变量。

```
if lode == 'bending'
  cload = 1;
else
  cload = 0.7;
end
if d <= 8
  csize = 1;
else
  csize = 1.189 * d^( - 0.097);
end
if temp <= 450
  ctemp = 1;
else
  ctemp = 1 - 0.0032 * (T - 840);
end
```

也可以将程序写为：

```
if lode == 'bending', cload = 1; else, cload = 0.7; end
if d <= 8, csize = 1; else, csize = 1.189 * d^( - 0.097); end
if temp <= 450, ctemp = 1; else, ctemp = 1 - 0.0032 * (Temp - 840); end
```

如第 1.8 节所述，同一行中的不同语句可用逗号或分号进行分隔。

4.2.3　for 循环

for 循环可按指定次数重复执行一系列语句，其常见结构为：

```
for 变量 = 表达式
  循环语句
end
```

这里循环语句可以是一条或多条，并且可以是变量的函数。

例 4.4 计算借款利息

已知借款总额为 L,借期 m 个月,年利率百分数为 I_a,求借款总利息。月还款额 P_{mon} 由下式得出:

$$P_{mon} = \frac{iL}{1-(1+i)^{-m}}$$

其中,

$$i = I_a/1200$$

是以十进制表示的月利率。每月偿还部分借款,偿还的钱中一部分用于支付利息,余下部分算做偿还借款。每月还贷后的余额称做收付差额。这些关系用数学表达式表示如下。设 $b_0 = L$,则:

$$\begin{aligned} i_n &= ib_{n-1} \\ P_n &= P_{mon} - i_n \qquad n = 1,2,3,\cdots,m \\ b_n &= b_{n-1} - P_n \end{aligned}$$

其中 i_n 是每月偿还利息 P_{mon} 的一部分,P_n 是每月偿还的本金,b_n 是未偿还的本金。则总利息为:

$$i_T = \sum_{j=1}^{m} i_j$$

计算总利息 i_T 的程序为:

```
loan = input('Enter loan amount:');                      ┐
durat = input('Enter term of loan in months:');          │ ←—输入
int = input('Enter annual interest rate:')/1 200;        ┘
ints = zeros(durat,1);                                   ┐
prins = ints;                                            │
bals = ints;                                             │ ←—初始化
pomn = (loan * int)/(1 - (1 + int)^( - durat));          │
bals(1) = loan;                                          ┘
for m = 2:durat + 1                                      ┐
  ints(m) = int * boal(m - 1);                           │
  prins(m) = pmon - ints(m);                             │ ←—计算
  bals(m) = bals(m - 1) - prins(m);                      │
end                                                      ┘
fprintf(1,'Total interest = $%8.2f\n',sum(ints))          ←—输出
```

正如第 1.2 节所述,该程序满足程序的一般结构:输入、初始化、计算和输出。本例的执行结果显示在 MATLAB 命令窗口中。程序执行结果为:

```
Enter loan amount:100000
Enter term of loan in months:360
Enter annual interest rate:8
Total interest = $164155.25
```

上面前三行是用户对程序中连续显示的 3 个提问的应答,用户在每个提问后输入数据,最后一行是程序结果。

MATLAB 建议在进入 for 循环和 for 嵌套循环体之前,利用子程序创建变量和数组。并利用 ze-

ros 函数指定数组大小。这将大大缩短这段程序的执行时间，否则 MATLAB 将在执行 for 循环体时自动动态分配内存，从而导致数据覆盖。

例 4.5 find 函数的实现

给定由正数和负数构成的任意长度向量 g，用程序实现下式：

```
indx = find(g > a)
```

这里 a 是由用户指定的一个值。本例中 $a=4$，向量 $g=[4\ 4\ 7\ 10\ -6\ 42\ 1\ 0]$。程序如下：

```
g = [4 4 7 10 -6 42 1 0];a = 4;
k = 0
indx = [ ];
for n = 1:length(g)
 if g(n) > a
  k = k + 1;
  indx(k) = n;
 end
end
disp(['Element locations for g(n) > 'num2str(a)':'num2str(indx)])
```

执行上述程序后，MATLAB 命令窗口显示：

```
Element locations for g(n) > 4.3 4 6
```

另一种生成 *indx* 的方法为：

```
g = [4 4 7 10 -6 42 1 0];a = 4;
k = 0;
indx = [ ];
for n = 1:length(g)
 if g(n) > a
  indx = [indx n];
 end
end
disp(['Element locations for g(n) > 'num2str(a)':'num2str(indx)])
```

例 4.6 cumsum 函数的实现

给定任意长度的向量 c，用程序实现：

```
Csum = cumsum(c)
```

用向量 $c=[\ 4\ 4\ 7\ 10\ -6\ 42\ 1\ 0]$ 来检验该程序。程序如下：

```
c = [4 4 7 10 -6 42 1 0];
Csum(1) = c(1)
for k = 2:length(c)
 Csum(k) = Csum(k - 1) + c(k);
end
disp(['Cumsum of c = 'num2str(Csum)])
```

执行上述程序后，MATLAB 窗口显示：

```
Cumsum of c = 4 8 15 19 61 62 62
```

另一种生成 *Csum* 的方法为：

```
c=[4 4 7 10 -6 42 1 0];
Csum(1)=c(1);
for k=2:length(c)
 Csum=[Csum Csum(k-1)+c(k)];
end
disp(['Cumsum of c='num2str(Csum)])
```

例 4.7　diag 函数的实现

给定($n \times n$)阶矩阵 $\boldsymbol{b}$，程序第一步实现：

```
v=diag(b)
```

第二步实现：

```
d=diag(v)
```

这里 v 是第一步得到的结果。用矩阵 $\boldsymbol{b}$=magic(4)检验程序，并假设不使用冒号和 zeros 函数。程序如下：

```
b=magic(4);[r c]=size(b);
for k=1:r
 v(k)=b(k,k);
end
disp(['Diagonal elements of b='])
disp(num2str(v))
for n=1:r
 for m=1:r
  if n==m
   d(n,m)=v(n);
  else
   d(n,m)=0.0;
  end
 end
end
disp(['Diagonal matrix d='])
disp(num2str(d))
```

执行完上述程序后，MATLAB 命令窗口显示：

```
Diagonal elements of b=
16  11  6  1
Diagonal matrix d=
16  0   0  0
0   11  0  0
0   0   6  0
0   0   0  1
```

如果允许使用冒号，则上述程序可简写为：

```
b=magic(4);[r c]=size(b);
```

```
for k = 1:r
 v(k) = b(k,k);
end
disp(['Diagonal elements of b = '])
disp(num2str(v))
d(1:r,1:c) = 0.0;
for n = 1:r
 d(n,n) = v(n);
end
disp(['Diagonal matrix d = '])
disp(num2str(d))
```

4.2.4 for 循环或 while 循环的提前终止

break 函数可实现 for 循环或 while 循环的提前终止。若 break 函数嵌套于 for 循环或 while 循环的内部，则返回到上一级的 for 循环或 while 循环。考虑下列程序段：

```
for j = 1:14
  ⋮
  b = 1
  while b<25
   ⋮
   if n<0
    break
   end
   ⋮
  end   <-----
  ⋮
end
```

当 $n<0$ 时，while循环将退出并从这个 end 语句之后继续执行。

由于某一特定条件未满足而终止的程序（或函数，见第 5 章），可使用 error 函数。error 函数常用来保证程序使用正确的数据并产生有意义的结果。当执行 error 函数时，将在 MATLAB 命令窗口中显示 error 函数引号内包含的内容，然后结束程序的执行过程。error 函数可在任意程序和函数中使用，而不只局限于 for 循环或 while 循环结构之中。

4.2.5 switch 语句

通常，switch 的语法结构为：

```
switch switch_expression
  case case_expression #1
     statements #1
  case case_expression #2
     statements #2
      ⋮
  case case_expression #n
     statements #n
  otherwise
     statements #n+1
end
```

当 *case_expression* #j = *switch_expression* ($j = 1,2,\cdots,n$)成立时，执行 *statements* #j 语句。

只有一个 case 被执行。执行完 *statements* #*j* 后,将执行 end 之后的语句。如果所有 *case _ expression* 都不满足,则执行 *statements* # $n+1$ 语句。当然 otherwise 语句不是必需的。若 otherwise 不存在且所有 *case _ expression* #*j* 都不满足,则执行 end 后面的语句。

显然,switch 结构与一系列 if-elseif-else-end 结构的功能相同。下面所述的 switch 结构阐述了这种执行方式。变量 k 已赋值或已出现过。

```
a = 3;
switchk
 case1
  disp('Case 1')          ←—仅当 k = 1 时执行。
 case{2,3}
  disp('Case 2 or 3')     ←—仅当 k = 2 或 3 时执行。
 case a^2
  disp('Case 9')          ←—仅当 k = 9 时执行。
 otherwise
  disp('Otherwise')       ←—仅当 k≠1、2、3 或 9 时执行。
end
```

例 4.8 作曲面的 4 个视图

在图形窗口的 4 个独立区域中分别绘制曲面 $z(x,y)$的 4 个视图,程序如下。其中 subplot 函数将窗口划分为网格,这里划分为 2×2 网格。变量 k 用来指定图像位于哪个网格区域内:$k=1$,左上角;$k=2$,右上角;$k=3$,左下角;$k=4$,右下角(关于 subplot 或其他绘图函数参见第 6 章和第 7 章)。surf 函数用于绘制以 x 和 y 为变量的函数 z 的三维视图,view 函数用于设定角度,变量 x、y 和 z 已被赋值或已计算得出其值。

```
for k = 1:4
 subplot(2,2,k)
 surf(x,y,z)
 switch k
  case 1
   view ( - 37.5,30)
  case 2
   view( - 70,30)
  case 3
   view( - 37.5,50)
  case 4
   view( - 70,50)
 end
end
```

4.3 程序控制结构的两项应用

现在给出两个综合利用程序控制结构的例子。

4.3.1 产生 2^k的析因表

有时经常要设计一些受多个因素影响的实验,要求程序分析每个因素及其所有组合对实验的

影响。这样的实验设计又叫做析因实验。如果实验中包含 k 个因素,每个因素对实验的影响为两个层面,那么就称该实验为 2^k 析因设计。将因素对实验的高水平影响记为"+",低水平影响记为"-"。$k=2$ 时,因素记为 A 和 B;$k=3$ 时,记为 A、B 和 C;$k=4$ 时,记为 A、B、C 和 D。

观察表 4.2 中的各列。符号"+"和"-"分别代表"+1"和"-1"。标号为 A、B、C 和 D 的列为主要因素对实验影响的数值,其余各列为交互效应对实验影响的数值,后者数值是各个主要因素对实验影响的数值的乘积。因此,ABC 列的值为 A、B 和 C 三列值的乘积,或者 AB 与 C 的乘积,或 A 与 BC 的乘积。例如,在第 7 行中($m=7$),$A=-1$,$B=+1$,$C=+1$,那么 ABC 为 $-1[=(-1)(+1)(+1)]$。另外,2^2 实验结果为前 3 列与前 4 行区域内的值;2^3 实验结果为前 7 列与前 8 行区域内的值;2^4 实验结果为前 15 列与前 16 行区域内的值。需注意的是当因素的个数增加时,前 $2^{(k-1)}$ 行($k>2$)被重复使用,以构成表 4.2 的第 $2^{(k-1)}$ 行下面的内容。

表 4.2 中的数据可反映最基本的 2^k 析因结果(参见第 14.6.2 节)。下面用程序产生 $k=2$、3 或 4 时析因表的对应值。程序如下:

```
k = input('Enter number of factors( = 2,3,or 4)');
s = ones(2^k,2^k - 1);
for r = 1:2:2^k
 s(r,1) = -1;
end
for c = 2:k
 e = 2^(c - 1);
 for r = 1:e
  s(r,e) = -1,
 end
for r = e + 1:2^(k)
  s(r,2^(c - 2)) = s(r - e,2^(c - 2));
 end
end
for m = 2:k
 e = 2^(m - 1);
 for j = 1:e - 1
  s(:,e + j) = s(:,j). * s(:,e);
 end
end
disp(s)
```

执行完上述程序后,显示如下:

```
Enter number of factors( = 2,3,or 4)3
  -1  -1   1  -1   1   1  -1
   1  -1  -1  -1  -1   1   1
  -1   1  -1  -1   1  -1   1
   1   1   1  -1  -1  -1  -1
  -1  -1   1   1  -1  -1   1
   1  -1  -1   1   1  -1  -1
  -1   1  -1   1  -1   1  -1
```

1　1　1　1　1　1　1

第一行的最后一个数“3”是由用户输入的。

表 4.2　2^k 析因符号

因素及其交互作用															
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
$k=2$			$k=3$				$k=4$								
A	B	AB	C	AC	BC	ABC	D	AD	BD	ABD	CD	ACD	BCD	ABCD	m
−	−	+	−	+	+	−	−	+	+	−	+	−	−	+	1
+	−	−	−	−	+	+	−	−	+	+	+	+	−	−	2
−	+	−	−	+	−	+	−	+	−	+	+	−	+	−	3
+	+	+	−	−	−	−	−	−	−	−	+	+	+	+	$4=2^2$
−	−	+	+	−	−	+	−	+	+	−	−	+	+	−	5
+	−	−	+	+	−	−	−	−	+	+	−	−	+	+	6
−	+	−	+	−	+	−	−	+	−	+	−	+	−	+	7
+	+	+	+	+	+	+	−	−	−	−	−	−	−	−	$8=2^3$
−	−	+	−	+	+	−	+	+	−	+	−	+	+	−	9
+	−	−	−	−	+	+	+	−	−	−	−	−	+	+	10
−	+	−	−	+	−	+	+	+	+	−	−	+	−	+	11
+	+	+	−	−	−	−	+	−	+	+	−	−	−	−	12
−	−	+	+	−	−	+	+	+	−	+	+	−	−	+	13
+	−	−	+	+	−	−	+	−	−	−	+	+	−	−	14
−	+	−	+	−	+	−	+	+	+	−	+	−	+	−	15
+	+	+	+	+	+	+	+	−	+	+	+	+	+	+	$16=2^4$

注意:如果 $z=s/2^{k/2}$,那么矩阵 z 是一个正交阵,即具有:

$$I = z'z$$

的性质(见练习 2.11)。这里,I 是一个(7×7)阶的单位阵。

4.3.2　用区域对分法求多个根①

MATLAB 的 *fzero* 函数可求取使 f(x)接近 0 的 x 值,即 *fzero* 可用于估计根的区域(参见第 5.5.1节)。但是,该函数一次只能求得一个根,要求得其他的根必须要重复调用该函数。

假设要自动求得使 f(x)=0 的一系列 x(x1,x2,…)的正值。当然,f(x)有多个根且随 x 的增加 f(x)的值也将改变。

自变量区域对分法是一种方程求根方法②。该方法的求根原理见图 4.1。给独立变量 x 赋初值 $x=x_{start}$,可从图中直接看出 $f(x_{start})$的符号。然后变量 x 加增量 Δ,$f(x_{start}+\Delta)$的符号也可确定。比较这两个值的符号,若符号相同(图 4.1 即如此),x 再加 Δ,判定 $f(x_{start}+2\Delta)$的符号并将它与 $f(x_{start})$的符号比较。若符号不相同,则 x 的值减少间隔的一半,即 Δ/2。从图 4.1

① 参见 S C Chapra 和 R P Canale, *Numerical Methods for Engineers*,第二版, McGraw-Hill,New York,1988 年,第 128ff 页。

② 尽管这种方法可以求出根,但它并不是最佳的方法。对于在 fzero 中所使用的方法,要想在给定精度上找到根,需要 2~3 次的作用。

中可以看出，当 $x = x_{start} + 3\Delta$ 时，f(x)的符号改变了，则下一个 x 为 $x = x_{start} + 5\Delta/2$。再将 $f(x_{start} + 5\Delta/2)$ 的符号与 $f(x_{start})$ 的符号比较。如果两个符号相同，则 x 增加当前间隔的一半，否则 x 减少间隔的一半。在本例中，$f(x_{start})$ 与 $f(x_{start} + 5\Delta/2)$ 符号相同，则下一个 x 值为 $x_{start} + 11\Delta/4$。重复以上过程，直到当前的增加量与当前的 x 的比值小于某一限值，即 $\Delta_{current}/x_{current} < t_0$。当满足这一限值标准时，根 $x_j = x_{current}$。重复求根过程，直到满足所要求的根的个数为止。求得一个根 x_j 后，设 $x_{start} = 1.05x_j$，将 Δ 设为最初值，重复求根过程。

写一段程序，采用区域对分法求满足下式的前 5 个根：

$$f(x) = \cos(ax) \approx 0$$

这里 a 是一个常数。当 x_j 增加的量与 x_j 的比值小于 $t_0 = 10^{-6}$ 时，x_j 是所求的根，并满足 $x_j \geq x_{start} \geq 0$，且 j = 1,2,3,…,n。因此，求根程序的输入参数为 n、x_{start}、t_0 和 Δ，对特定的 f(x)还有常数 a。

在第 5 章中，将把这段程序转换成一个函数，以便求解 f(x)的根，见图 4.1。$n = 5$，$t_0 = 10^{-6}$，$x_{start} = 0.2$，$\Delta = 0.3$，$a = \pi$。程序如下：

```
n = 5; a = pi; increment = 0.3; tolerance = 1e-6; xstart = 0.2;
x = xstart;
dx = increment;
for m = 1:n
 s1 = sign(cos(a * x));
 while dx/x > tolerance
  if s1 ~= sign(cos(a * (x + dx)))
   dx = dx/2;
  else
   x = x + dx;
  end
 end
 route(m) = x;
 dx = increment;
 x = 1.05 * x;
end
disp(route)
```

执行结果为：

```
0.5000    1.5000    2.5000    3.5000    4.5000
```

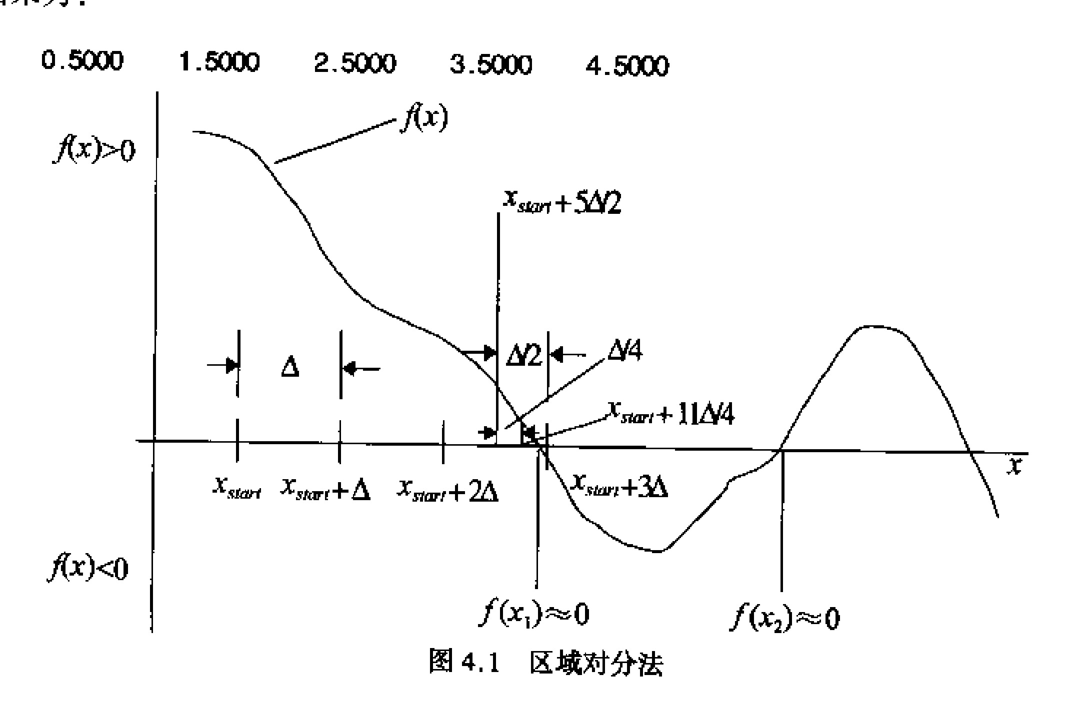


图 4.1　区域对分法

练习

4.1 编写程序完成第 4.1 节中提到的逻辑运算,即给定任意长度的向量 $\boldsymbol{h}$,判定其元素值大于 a 和小于 b 时结果是否为 1,并将结果用向量 $\boldsymbol{v}$ 表示出来。设 $\boldsymbol{h}=[1\ 3\ 6\ -7\ -45\ 12\ 17\ 9]$, $a=3$, $b=13$,检验所设计的程序。

[答案:$\boldsymbol{v}=[0\ 0\ 1\ 0\ 0\ 1\ 0\ 1]$。]

4.2 估计由下式描述的 n 个采样值 x_j 的方差:

$$s_n^2=\frac{1}{n-1}\left[\sum_{j=1}^{n}x_j^2-n\bar{x}_n^2\right]\qquad n>1$$

其中,

$$\bar{x}_n=\frac{1}{n}\sum_{j=1}^{n}x_j$$

这是平均值。方差由 var 决定。

编写计算 $n>1$, x=[45 38 47 41 35 43]时样本方差 s_n^2 的程序。

[答案:[24.5000,22.3333,16.2500,24.2000,19.9000]。]

4.3 给定 $a>0$,假设任意初值 $x_0>0$,下面的关系式将限定数$\sqrt{a}$ 在容限为 t_0以内:

$$x_{n+1}=\frac{1}{2}\left(x_n+\frac{a}{x_n}\right)\qquad n=0,1,2,\cdots$$

其中 $x_{n+1}\approx\sqrt{a}$(当 $x_0<0$ 时,有负方根)。编写程序,当 $a>7$ 时,在 $|x_n-x_{n+1}|<10^{-6}$条件下确定$\sqrt{a}$ 在$(a)x_0=3$ 和$(b)x_0=100$ 时,计算将重复多少次。第一次重复是确定 x_1。[提示:上述关系式不是 n 的显式函数。这里下标 n 标明下一值 x_{n+1}是当前值 x_n的函数。因此,每次循环后新旧值都被保存,n 记录循环次数。]

[答案:$(a)n_{\text{iterations}}=4$, $(b)n_{\text{iterations}}=10$。]

4.4 考虑下式:

$$x_{n+1}=x_n^2+0.25\qquad n=0,1,2\cdots,N$$

对 $x_0=0$,编写两段程序求 $n=0,5,10,\cdots,200$ 时 x_n 的值。第 1 段程序采用 for 循环结构,第 2 段程序采用 while 循环结构。x_n 收敛为何值? plot 函数的第 3 个参数设为 plot(…, …, 'ks'),画出 x_n 的平方曲线图。x 轴代表 n,y 轴为 x_n。注意,对 $n=0,1,2,\cdots,200$ 的所有 x_n值都要计算出来,但画图时只须每隔 5 点画出即可。另外,与练习 4.3 不同的是,x_n必须以向量形式保存,以便随时显示出其特定元素。

4.5 X^2统计用于拟合优度的检验(见练习 14.6)。它定义为:

$$X^2=\sum_{i=1}^{k}\frac{(\boldsymbol{x}_i-\boldsymbol{e}_i)^2}{\boldsymbol{e}_i}$$

其中 $\boldsymbol{e}_i$和$\boldsymbol{x}_i$是长度为 k 的独立向量。

若 $\boldsymbol{e}_i<5$,将 $\boldsymbol{e}_i$, $\boldsymbol{x}_i$分别与$\boldsymbol{e}_{i+1}$, $\boldsymbol{x}_{i+1}$相加。如果 $\boldsymbol{e}_i+\boldsymbol{e}_{i+1}$的和仍然小于 5,再将 $\boldsymbol{e}_{i+2}$与 $\boldsymbol{e}_i+\boldsymbol{e}_{i+1}$相加。这一过程一直重复到和大于等于 5。如 $\boldsymbol{e}_i\geq5$ 且剩余值 $\boldsymbol{e}_{i+1}$, $\boldsymbol{e}_{i+2}$, $\cdots$, $\boldsymbol{e}_k$的和小于 5,则将这些剩余值与 $\boldsymbol{e}_i$相加。

编写程序在下列条件下计算 X^2的值。并用下列向量检验计算结果，这些向量代表了 3 种不同的情况：

(i) $\boldsymbol{x}$ = [1 7 8 6 5 7 3 5 4]且 $\boldsymbol{e}$ = [2 6 10 4 3 6 1 2 3]；

(ii) $\boldsymbol{x}$ = [7 11 13 6]且 $\boldsymbol{e}$ = [6 10 15 7]；

(iii) $\boldsymbol{x}$ = [3 14 20 25 14 6 2 0 1 0]且 $\boldsymbol{e}$ = [4 12 19 19 14 8 4 2 1 1]。

保存程序用于练习 14.6(b)。提示：通过检验 cumsum(e)的元素，可获得最为简洁的程序，随着计算过程的逐步深入，e 的长度将有所变化。

[答案：

(i) $\boldsymbol{e}_{\text{modified}}$ = [8 10 7 6 6]，$\boldsymbol{x}_{\text{modified}}$ = [8 8 11 7 12]，X^2 = 8.852 4；

(ii) $\boldsymbol{e}_{\text{modified}}$ = [6 10 15 7]，$\boldsymbol{x}_{\text{modified}}$ = [7 11 13 6]，X^2 = 0.676 2；

(iii) $\boldsymbol{e}_{\text{modified}}$ = [16 19 19 14 8 8]，$\boldsymbol{x}_{\text{modified}}$ = [17 20 25 14 6 3]，X^2 = 5.634 9。]

4.6 给定两个多项式：

$$y(x) = p_1 x^n + p_2 x^{n-1} + \cdots + p_n x + p_{n+1}$$
$$z(x) = s_1 x^m + s_2 x^{m-1} + \cdots + s_m x + s_{m+1}$$

编写程序求和，即分别在 $m = n$，$m < n$ 和 $m > n$ 的情况下求 $h(x) = y(x) + z(x)$。多项式的求和是相同指数的系数求和。假设程序输入为向量 $\boldsymbol{p} = [p_1\ p_2 \ldots p_n\ p_{n+1}]$ 和 $\boldsymbol{s} = [s_1\ s_2 \ldots s_m\ s_{m+1}]$。

用下列数组检查所编写的程序：

(i) $\boldsymbol{p}$ = [1 2 3 4]且 $\boldsymbol{s}$ = [10 20 30 40]；

(ii) $\boldsymbol{p}$ = [11 12 13 14]且 $\boldsymbol{s}$ = [101 102]；

(iii) $\boldsymbol{p}$ = [43 54 55]且 $\boldsymbol{s}$ = [77 66 88 44 33]。

[答案：(i) h = [11 22 33 44]；(ii) h = [11 12 114 116]；(iii) h = [77 66 131 98 88]。]

4.7 编写程序计算 1999 和 2000 年中的任意一天是星期几并计算出是一年中的第几天。输入格式为：month/day/year – xx/xx/xxxx。程序输出结果以下列方式显示：

```
The date 5/31/2000 is the 152 day of the year and falls on a Wednesday.
```

可利用 str2num、deblank 和 findstr 函数进行编程。

第5章　函　　数

本章论述了 MATLAB 函数的创建及其不同用法，并介绍了几个常用于求解工程问题数值解的内置函数。

5.1　概述

程序文件是一种 m 文件，而函数文件是另一类 m 文件。在 MATLAB 中，函数文件是能够创建局部独立空间的程序文件。在函数内定义的所有变量名对函数而言是局部的，既不影响其他程序文件和函数文件中同样的变量名又不被其所影响。所有的 MATLAB 内置函数都是这种类型。一个函数文件的第一个非注释行必须按照特定的格式书写，这将在第 5.2 节中给出说明。典型的情况是，用户创建的 MATLAB 程序由一程序文件组成，并且调用了任意多个用户自定义函数和 MATLAB 内置函数。

5.1.1　为什么使用函数

MATLAB 的几个内置函数需要创建一些函数，除此之外，创建函数还有以下几个原因（见第 5.5 节）。它们包括：

1. 避免代码复制。
2. 限制因改变特定代码段而带来的影响。
3. 提高代码重用性。
4. 减少整体代码的复杂性，增加可读性和可管理性。
5. 隔离复杂操作。
6. 提高移植性。
7. 更易于调试和隔离错误。
8. 因为每一段程序被“最优化”，所以可提高整体性能。

通过函数对程序的划分可以减少对多余数据的使用，因为对每个函数而言，提供的数据必须是已知的。

5.1.2　函数的命名

函数应该选择与程序功能有关系的名字。函数名典型长度在 9 至 20 个字符之间，并且应该使用标准的或一致的规则。例如，所有的程序文件名以 *scr* 开始，但所有的函数名没有这一前缀。函数名的正确选择可有效减少函数内的注释行。可以参考第 1.3 节建议的命名规则。

5.1.3　函数的长度

函数的代码长度从两行到几百行不等。然而，函数的长度应该由函数内部功能的粘合性

所支配——也就是说，在某种程度上，它做一件事情而不是其他任何事情。例如，sin(*x*)是100%粘合的，而函数 *sincostan*(*x*)则不然，它由 sin、cos 和 tangent 组成，粘合性较低，因为这三部分相互独立，互不相关。可利用多个高粘合性函数创建一个粘合性的函数，这样做的好处是具有可靠性——低差错率。当函数的粘合程度较低时，要隔离错误很困难。

5.1.4 函数的调试

在函数(和程序)创建过程中，应该独立检验每行代码，以确保写完几行后即可正确工作。MATLAB 特别适合于这一处理过程，它简单地忽略每一行的分号。进一步说，除了那些使用大规模向量和矩阵或迭代求解的程序外，忽略分号可节省大量时间。应该用一些独立的计算或估计方法进行检验。在检验/调试阶段，应当把提供中间输出的代码行注释掉，而不是删除掉，等到整个函数被检验能正常工作时再恢复。函数被检验正确后，如果必要，应尽量减少它的执行时间。然而，创建可正确执行的程序是基本目标。

5.2 函数文件

一个函数至少有两行代码，第一行是 MATLAB 所要求的格式，在函数程序中没有结束字符或表达式。例如 end 语句，它被用于 for、while、if 和 switch 结构而不用于函数。而且，m 文件的名字应该和函数名字相同，扩展名为.m。

引入和导出函数的变量数目和它们的类型(标量、向量、矩阵)由函数接口控制，函数接口是函数程序的第一个非注释行。接口的一般形式如下：

```
function OutputVariable = FunctionName(InputVariables)
% comments
expression(s)
```

MATLAB 用紧跟函数接口语句之后的注释创建此函数的帮助信息——当在 MATLAB 命令行中输入 Help 函数名时，所有相邻的注释行都将出现在 MATLAB 命令窗口中。任何出现在 function 语句之前的注释将不作为帮助文件的部分。

表达式的写法和程序文件的写法相同。函数所使用的输入变量用变量名定义，其他变量由表达式定义。输入和输出变量可以是标量、向量或矩阵，也可以由数值或字符串组成。可以有任意多个输入变量，变量间用逗号隔开。如果有三个输入变量 a、b 和 c，程序的接口语句为：

```
function OutputVariable = FunctionName(a,b,c)
```

正确创建函数必须要明白几个概念。第一，函数中使用的变量名不必和从 MATLAB 命令窗口、程序文件或另一个函数中调用此函数时使用的名字相一致。但是圆括号内输入变量的位置确定了信息的传递顺序——在调用语句中的第一个参数将值传递到函数接口定义中的第一个参数，依此类推。被分配的名字应当被看做占位符。

第二，每个参数名对函数而言都是局部的，并且仅在函数范围内有意义。同样的名字可以在调用这一函数的程序文件中使用，也可在此函数调用的另一个函数中使用。然而，函数语句和调用语句中变量名必须具有相同类型，可以是标量、向量、矩阵或字符串的一种，这样可使函数的表达式按预先设定的类型工作。例如，如果变量没有正确的阶数，两个行向量的乘法会导

致一错误信息。进而,函数语句的输入变量名相当于它们出现在等号的左边。在上例中,输入变量(a,b,c)等同于语句:$a=\cdots;b=\cdots;c=\cdots$。输出变量可以有几种形式,每种形式对函数语句的返回值都有影响,这些将在下面讨论。

函数文件可被存储在任何已被定义或将被定义的路径下,文件名为"*FunctionName.m*"。为说明函数的创建过程,将讨论一个例子和它所具有的三种形式。

5.2.1 形式#1:输入参数单独标识和单个输出变量

以下是函数中计算的等式:

$$x=\cos(at)+b$$

$$y=|x|+c$$

函数返回 x 和 y 的值。现在创建一个 *ComputeXY* 函数计算这些量①,文件名为 *ComputeXY.m*。

```
function zanswer = ComputeXY(t,a,b,c)
% Computation of-
%   x = cos(at) + b
%   y = |x| + c
%Scalars:a,b,c
%Vector:t
%Matrix:zanswer
x = cos(a * t) + b;
zanswer = [x;abs(x) + c];
```

在 MATLAB 命令窗口中输入:

```
help ComputeXY
```

则显示如下内容:

```
Computation of-
  x = cos(at) + b
  y = |x| + c
Scalars:a,b,c
Vector:t
Matrix:zanswer
```

MATLAB 通过函数名区分函数,而不是通过函数输入和输出变量的特征,因此必须根据函数内的使用方法确保输入和输出变量的数目和类型正确。为了响应 help 请求,函数的注释中应指明这些限制。这里,假定 t 是一标量或是一向量,a,b,c 是标量。因此,当 t 是长度为 n_t 的向量时,函数返回($2\times n_t$)阶矩阵 *zanswer*,且 *zanswer*$(1,:)=x(:)$,*zanswer*$(2,:)=y(:)$。

现在来看程序文件中调用此函数的代码部分,一种调用情况为:

```
vick = ComputeXY(0:pi/4:pi,1.4,2,0.75);
```

---

① 本例中的注释表明了其用途。在本书给出的大量程序和函数中,为了提高清单的可读性,注释行均被省略。大多数情况下,程序中重要的部分在程序和函数的相关内容加以讨论,也可以从上下文中看到。

在圆括号中的位置区分参数，对此函数而言，$t=0,\pi/4,\pi/2,3\pi/4$ 和 π，$a=1.4$，$b=2$，$c=0.75$。执行这一语句，得到如下结果：

```
vick(1,:)→[3.0000  2.4540  1.4122  1.0123  1.6910]
vick(2,:)→[3.7500  3.2040  2.1622  1.7623  2.4410]
```

其中 $x=vick(1,:)$，$y=vick(2,:)$。如果函数 *zanswer* 被写为：

```
zanswer = [x abs(x) + c];
```

则 *ComputeXY* 执行后将返回$(1\times 2n_t)$阶向量：

```
vick→[3.0000  2.4540  1.4122  1.0123  1.6910  3.7500  3.2040  2.1622  1.7623  2.4410]
```

其中，$x=vick(1:5)$，$y=vick(6:10)$。

5.2.2 形式#2：多个参数作为一个向量输入和单个输出变量

函数清单

```
function zanswer = ComputeXY(t,w)
x = cos(w(1) * t) + w(2);
zanswer = [x;abs(x) + w(3)];
```

调用格式

```
vick = ComputeXY(0:pi/4:pi,[1.4 2 0.75]);
```

通过向量 w 中各元素的位置，可确定出输入参数：$w(1)=a=1.4$；$w(2)=b=2$；$w(3)=c=0.75$，其中 a,b,c 对应于形式#1中定义的三个量。

5.2.3 形式#3：多个参数作为一个向量输入和多个输出变量单独标识

函数清单

```
function [x,y] = ComputeXY(t,w)
x = cos(w(1) * t) + w(2);
y = abs(x) + w(3);
```

调用格式

```
[u,v] = ComputeXY(0:pi/4:pi,[1.4 2 0.75]);
```

在语句 function[x,y]中，逗号是必需的，但是在[u,v] = …中，逗号可用空格代替。在程序文件中，如果对 *ComputeXY* 的调用写做 $q=ComputeXY(\cdots)$，则只有第一个值赋给 q，即 q 将对应向量 $\boldsymbol{x}$。当执行下面的语句时：

```
[u,v] = ComputeXY(0:pi/4:pi,[1.4 2 .75]);
```

可得到如下结果：

```
u→[3.0000  2.4540  1.4122  1.0123  1.6910]
v→[3.7500  3.2040  2.1622  1.7623  2.4410]
```

而执行下面语句时：

```
q = ComputeXY(0:pi/4:pi,[1.4 2.75]);
```

得到如下结果：

```
q→[3.0000  2.4540  1.4122  1.0123  1.6910]
```

但在程序文件中将得不到向量 $\boldsymbol{y}$。

函数调用的左边变量形式必须与函数自身定义相一致。许多 MATLAB 内置函数利用这一

特性,使它的使用对上下文具有依赖性。例如,polyval 的 Help 文件。

因为函数定义中的参数只是数字值的占位符,所以函数执行时这些数字值常占用各自的位置。在调用语句中,可以适时插入一些正确的结构表达式。为阐明这一点,考虑如下具有形式 # 3 的函数程序:

```
n=4;
c=linspace(1,1.4,n);
for k=1:n
 [u,v]=ComputeXY(0:pi/4:pi,[c(k)sqrt(1.8/(1+k)^3)1/.85]);
    ⋮
```

最后,介绍一种情况,即函数的返回结果是一个向量,而且在程序中被重新定义为矩阵的一行。为简单起见,仅考虑 x 的值,这些值在 for 循环中返回。程序段如下:

```
n=4;
c=linspace(1,1.4,n);
p=zeros(4,n)
for k=1:4
 p(k,:)=ComputeXY(0:pi/4:pi,[c(k)sqrt(1.8/(1+k)^3)1/.85]);
          ⋮
```

因为 $k=1,2,3,4$ 且向量 t 的长度是 5,可看出在这种情况下,$p(=u=\cos(at)+b)$将是一个(4×5)阶矩阵。符号 $p(k,:)$指矩阵 p 的第 k 行,矩阵 p 中任何一行的所有元素与 *ComputeXY* 返回的行向量对应。

5.2.4 两种特殊情况

如果创建的函数仅仅用于在 MATLAB 命令窗口中绘图或仅仅将数据写到文件中,在这些情况下没有返回值。那么,函数的接口行可简写如下:

```
function FunctionName(InputVariables)
```

当函数仅仅用来以指定的方式存储数据时,函数不需要任何输入参数。在这种情况下,函数接口行的形式为:

```
function OutputVariables = FunctionName
```

或:

```
function [a,b,...] = FunctionName
```

一般情况下,只有执行完最后一条语句时,函数才返回结果。如果想提前返回,则用如下语句实现:

```
return
```

下面的实例给出了某函数的一部分:

```
function c = FunctionName(x,a,b)
 if length(x)==1|nargin ~=3
  c=x
  return
```

```
    end
    ⋮
```

nargin 给出了调用函数时输入变量的实际数目。

5.3 INLINE

在 MATLAB 命令窗口、程序或函数中创建局部函数时，可以使用 inline。优点是它不必存储为一单独文件。在运用中有几点限制：它不能调用另一个 inline 函数，它只能由一个 MATLAB 表达式组成，并且它只能返回一个变量——显然形式 # 3 不允许这样。因而，任何要求逻辑运算或乘法运算以求得最终结果的场合，都不能应用 inline。

inline 函数的一般形式为：

```
FunctionName = inline('expression','p1','p2',…)
```

其中，*expression* 本身是任何有效的 MATLAB 表达式，$p1,p2$…是出现在 *expression* 中的所有变量的名字。下面的例子用于说明 inline。创建一函数 *FofX* 求解下式：

$$f(x) = x^2\cos(ax) - b$$

其中 a 和 b 是标量。有：

```
FofX = inline('x.^2. * cos(a * x) - b','x','a','b')
```

显示在 MATLAB 命令窗口中。有：

```
FofX =
  Inline function:
  FofX(x,a,b) = x.^2. * cos(a * x) - b
```

如果用一分号结束表达式，那么显示将会被禁止。

因而，当在 MATLAB 命令窗口中输入如下表达式时：

```
g = FofX([pi/3 pi/3.5],4,1)
```

系统输出为 $g = [-1.5483\ \ -1.7259]$。

对函数而言，inline 的形式被用于很多 MATLAB 命令中，它要求先创建函数，然后再执行命令。几个应用实例在第 5.5 节给出。

5.4 使用 FEVAL(函数的函数)创建的函数

有许多 MATLAB 内置函数要求用户通过它们创建特定形式的用户自定义函数。第 5.5 节讨论了几个这样的函数。除此之外，有些情况下用户也想具备这种功能。MATLAB 用 feval 可以实现。

以第 4.3.2 节的结果为例解释这一过程，例子中的求根程序确定了某函数 $f(x)=0$ 的 m 个最接近的根。将程序重新定义为 *ManyZeros* 函数，它包含在 *ManyZeros.m* 文件中。函数 $f(x)$和它的名字可以是任意的。此外，还需要定义 $f(x)$能输入多个参数。求根程序要求 4 个输入：m 为所希望的最接近根的数目，x_s为根的初始值，t 确定接近零点$f(x_{root})$程度的计算误

差,Δ 为初始增量。

对于此例,定义:

$$f(x) = \cos(\beta x) - \alpha \qquad \alpha \leqslant 1$$

因而必须赋给用户定义的函数两个值:β 和 α。函数名为 *CosBeta*,它在文件 *CosBeta.m* 中。为实现这些函数,需要一程序文件。函数和程序如图 5.1 所示。根据 MATLAB 帮助文件,feval 的第一个参数是函数的名字,另外的参数是传递到函数的数值。函数 *ManyZeros* 要求下列输入变量:(1)定义 $f(x)$ 的函数名,即字符串变量 *zname*(='CosBeta');(2)6 个参数,前 4 个对应于 m、x_s、t 和 Δ,其余的 2 个是 β 和 α,这些被传递到函数 *CosBeta*。根据 m 的值,结果 c 可能为一标量($m=1$)或为一 $\boldsymbol{m}$ 阶向量。

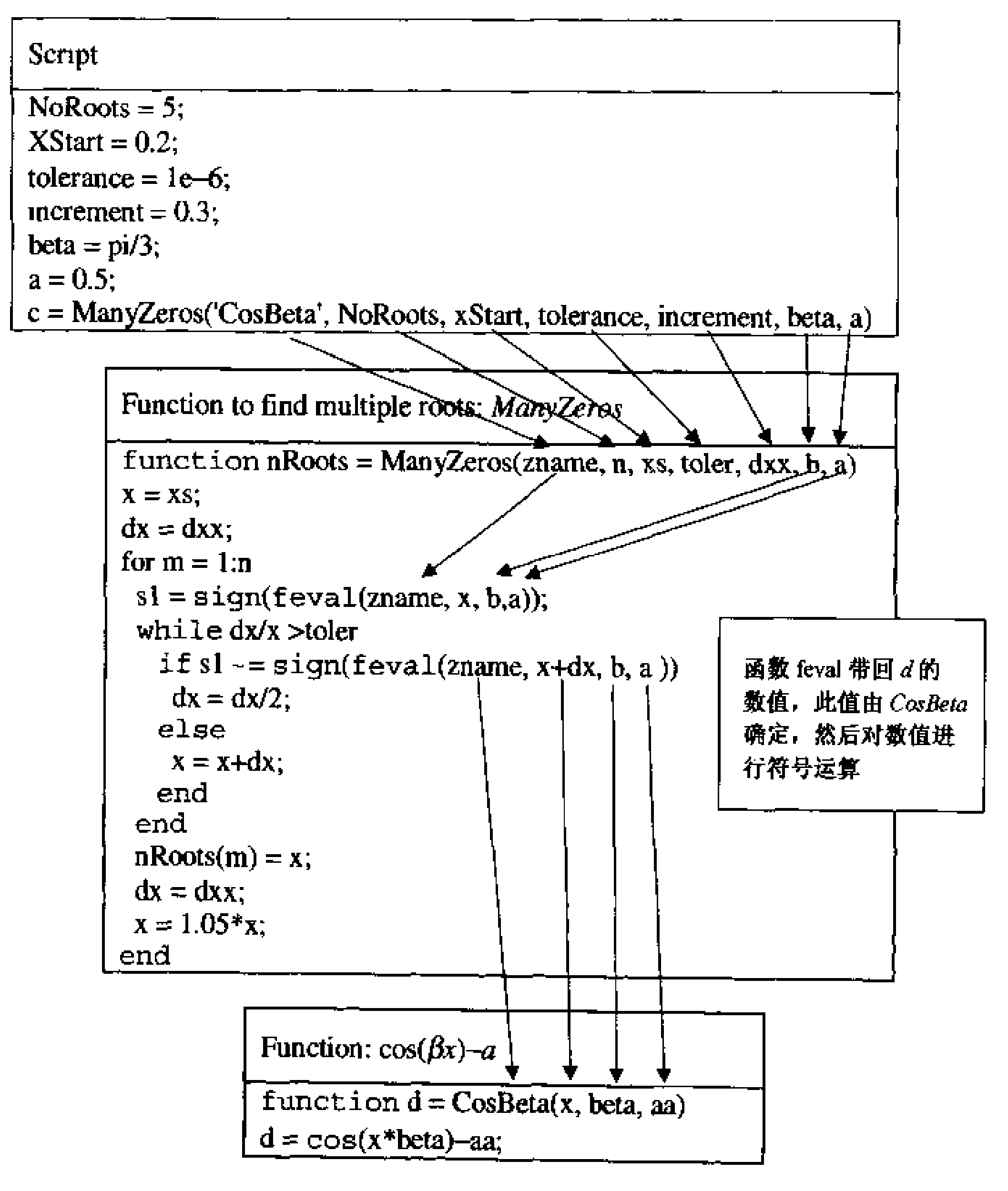


图 5.1 feval 在用户自定义函数中的应用图解

为访问函数 *CosBeta* 并返回其数值,使用 MATLAB 函数 feval。MATLAB 函数 sign 用来判断 feval 返回数值的符号。要注意,在程序文件中定义的函数输入变量名和两个函数文件中定义的可能完全不同。这说明,由于函数中的变量是局部的,所以起作用的是参数的位置和用途,

而不是名称的字符描述。

5.5 使用 FEVAL 的 MATLAB 函数

MATLAB 提供了几个函数用来计算用户定义函数，分别是：

```
fzero——找 f(x)=0 的根
roots——找一多项式的根
quad8——在给定间隔内对 f(x)的数值积分
trapz——在给定间隔内对 f(x)的数值积分
polyarea——决定一闭合曲线内的面积
fminbnd——在一特定间隔内找到 f(x)的局部最小值
ode45——常微分方程组的数值积分
fsolve——非线性方程组的数值解
```

当使用内置函数时，用户自定义函数的输入参数或输出结果应满足某种特定的要求。下面几节说明了这些不同的要求，并且在函数的帮助文件中清楚地表示出来。

5.5.1 零点函数——fzero 和 roots/poly

容许误差为 t_0时，在 x_0的相临区域内或在$[x_1\ x_2]$范围内，利用内置函数 fzero 可以求解 $f(x)=0$。它也可以把 p_j参数传递到函数定义$f(x)$中。一般的表达式为：

```
fzero(FunctionName,x0,options,p1,p2,…)
```

其中 *FunctionName* 可以是由单引号引用的函数文件名，但是没有后缀“.m”；也可以是不带引号的函数的变量名，该函数由 inline 创建。$x0=x_0$或 $x0=[x_1\ x_2]$，$p1$、$p2$ 等等，都是参数 p_j。此后用 optimset 设置 *options*。

内置函数 optimset 是一通用参数调整函数，它被优化工具箱中的 MATLAB 函数使用，最简单的例子是使用 optimset 关闭显示，如下所示：

```
options = optimset('display','off');
```

上式将会用在每个 fzero 语句之前。通过 optimset 的帮助文件可知它的属性会根据函数变化而变化。

函数接口具有如下形式：

```
function z = FunctionName(x,p1,p2,…)
```

其中 x 是独立变量，为了找到使 $f(x=z)\approx 0$ 的值，fzero 不断改变 x 的值。独立变量必须总出现在这一位置。用 MATLAB 创建的大多数用于计算的函数应满足这一要求，本章所述的函数也都这样。下面介绍 fzero 的使用。

函数 $f(x)$可以是 MATLAB 函数也可以是用户创建的函数。假定要求 $\cos(x)$在 $x=6$ 附近的根，写出如下语句：

```
options = optimset('display','off');
w = fzero('cos',2*pi,options)/pi
```

结果 $w=1.5000$，也就是 $\cos(1.5\pi)=0$。然而，

```
options = optimset('display','off');
w = fzero('cos',2.04 * pi,options)/pi
```

则结果 $w = 2.5000$，如果：

```
options = optimset('display','off');
w = fzero('cos',2.03 * pi,options)/pi
```

则 $w = 1.5000$。

对于多值函数，应使用 $x0 = [x_1\ x_2]$形式，并且准确划定范围。然而，如果 $f(x_1)$的符号和 $f(x_2)$的符号相同，则会得到一错误结果。所以，下式将会导致一个错误：

```
options = optimset('display','off');
w = fzero('cos',[0 2 * pi],options)/pi
```

而式：

```
options = optimset('display','off');
w = fzero('cos',[0.6 * pi 2 * pi],options)/pi
```

得出 $w = 1.5000$。对多值函数而言，其特征是不分先后，所以应当首先确定根的大概位置。

另一方面，如果对方程 $J_1(x) = 0$(其中 $J_1(x)$是一类 1 阶贝塞尔函数[①])在 3 附近的根感兴趣，则下述语句将得不到预想的结果。

```
options = optimset('display','off');
w = fzero('besselj',3,options)
```

因为 besselj 的参数是 besselj(n, x)，其中 n 是阶次(本例为 1)，x 是独立变量。因而第一个参数是 n 而不是 x。所以必须创建一个新的函数：

```
function v = besseljx(x,n)
v = besselj(n,x);
```

然后调用 fzero 函数：

```
options = optimset('display','off');
a = fzero('besseljx',3,options,1)
```

它返回正确的结果 $a = 3.8317$。注意，为把参数 $p_1 = 1$ 传递给函数 *besseljx*，在 fzero 的第四个位置必须赋以数值 1。

一个两端受力的梁，其固有频率系数由下式确定：

$$f(x) = \cos(x)\cosh(x) - 1$$

首先创建一名为 *ccbeam* 的函数：

```
function s = ccbeam(x)
s = cos(x) * cosh(x) - 1;
```

那么下式将返回 $q = 4.7300$：

```
options = optimset('display','off');
```

① 参见：F B Hildebrand, *Advanced Calculus for Applications*, Prentice-Hall, Saddle River, NJ, 1976 年。

```
q = fzero('ccbeam',4,options)
```

函数 $f(x)$如图 6.8 和图 6.9 所示。

也可以按下列方式之一使用 inline 求解方程的根。

方式 # 1

```
qcc = inline('cos(x) * cosh(x) - 1','x');
options = optimset('display','off');
q = fzero(qcc,4,options)
```

方式 # 2

```
options = optimset('display','off');
q = fzero(inline('cos(x) * cosh(x) - 1','x'),4,options)
```

前面提到,当 *qcc* 由 inline 函数定义时,它没有单引号。这是所有 MATLAB 内置函数对用户自定义函数的要求。当函数在程序中使用不只一次时,必须使用方式 # 1。因为在方式 # 2 中,函数在被定义之前,对程序中其他表达式是无效的,所以相比之下,方式 # 1 更可取。

现在确定下式的最小的 5 个根(不包括 $x = 0$)

$$f(x) = \cos(x)\cosh(x) - 1$$

该过程首先在 x 范围内画出函数,以观察在哪个位置上的 x 使 $f(x) = 0$。利用这些信息可以得到 5 个范围,fzero 可在每个范围上得到一个根。程序如下:

```
qcc = inline('cos(x). * cosh(x) - 1','x');
options = optimset('display','off');
%x = linspace(0,20);
%plot(x,qcc(x))
%axis([0 20 -10 10])
xo = [3 5];
for n = 1:5
 q(n) = fzero(qcc,xo,options);
 xo = [1.05 * q(n) q(n) + 4];
end
disp(['Lowest five roots are:'num2str(q)])
```

执行时,MATLAB 命令窗口显示如下内容:

```
Lowest five roots are:4.73004 7.8523 10.9956 14.1372 17.2788
```

为易于绘出 *qcc* 函数,在其定义中使用点操作。fzero 不要求这些点操作。因为 *qcc* 在很大的正负范围内变化,用 axis 限制显示值极大地提高了图形的求解能力。在这种情况下,限制 y 轴为 ± 10。见第 6.2 节,从图中可以确定,x 的搜索范围应从前一个根的 1.05 倍处开始,到前一个根加 4 的位置,搜索范围的上限应略高于下一个零点。在程序的实际编制过程中,先让前 5 个表达式执行完毕,并对结果进行分析,然后再写出后 6 个表达式,一旦确认了这些信息,就可以将表达式注释掉。

如果 $f(x)$不是一常数,本例为 4,而是要求有几个不同的值 C_j, $j = 1,2,3,4$。那么上面的程序将变成如下形式:

```
qcc = inline('cos(x). * cosh(x) - 1','x');
options = optimset('display','off')
```

```
C=[4 5 6 5 0];
xo=[3 5];
for n=1:5
 q(n)=fzero(qcc,xo,options);
 xo=[1.05*q(n) q(n)+C(n)];
end
disp(['Lowest five roots are:'num2str(q)])
```

由于向量 C 的下标要求,所以必须增加第 5 个元素 0,但 fzero 并不使用它。

最后,求满足下式的 a 值:

$$\sum_{j=1}^{1000} \frac{1}{j^2 - a} = 0$$

并在 MATLAB 窗口中用 8 位数字显示结果。首先创建一个名为 *suma* 的函数:

```
function z=suma(a)
z=sum(1./([1:1000].^2-a));
```

初始值设为 $\pi/2$,语句如下:

```
options=optimset('display','off');
fofa=fzero('suma',pi/2,options);
disp(['The value of a is 'num2str(fofa,8)'.'])
```

执行这些语句,可得如下结果:

```
The value of a is 2.0465776.
```

若用 inline 求解 a,则写出下面形式的语句:

```
options=optimset('display','off');
fofa=fzero(inline('sum(1./([1:1000].^2-a))','a'),pi/2,options);
disp(['The value of a is 'num2str(fofa,8)'.'])
```

当 $f(x)$ 是具有下面形式的多项式时:

$$f(x) = c_1x^n + c_2x^{n-1} + \cdots + c_nx + c_{n+1}$$

用下面的函数可以很容易找到它的根:

```
roots(c)
```

其中 $c=[c_1\ c_2\cdots c_{n+1}]$。例如,若:

$$f(x) = x^4 - 10x^3 + 35x^2 - 50x + 24$$

那么,

```
r=roots([1 -10 35 -50 24])
```

给定 $r=[4\ 3\ 2\ 1]'$。根的逆为:

```
c=poly(rts)
```

上式返回值 c 是多项式的系数,*rts* 是根向量。一般情况下,*rts* 可以是实向量,也可以是复向量。

多项式可以用下式实现乘法运算:

```
conv(a,b)
```

其中 a 和 b 是包含某个多项式系数的向量。一多项式如下：

$$g(x) = x^2 - 4$$

则乘积 $h(x) = g(x)f(x)$ 通过下式得到：

```
h = conv([1 0 -4],[1 -10 35 -50 24])
```

其结果为：

```
h→[1 -10 31 -10 -116 200 -96]
```

相应的多项式为：

$$h(x) = x^6 - 10x^5 + 31x^4 - 10x^3 - 116x^2 + 200x - 96$$

这一多项式的根应当是 $f(x)$ 和 $g(x)$ 的根，即 ±2.0。运行如下语句：

```
rh = roots([1 -10 31 -10 -116 200 -96])
```

结果为：

```
rh→[4.0000 -2.0000 3.0000 2.0000 2.0000 1.0000]'
```

注意，根的结果按任意顺序排列。

5.5.2 数值积分函数——quad8、trapz 和 polyarea

内置函数 quad8 在容许误差 t_0 内，对 $f(x)$ 从下限 a 到上限 b 进行数值积分。它也可以把 p_j 参数传递到定义 $f(x)$ 的函数中去。quad8 的一般表达式为：[①]

```
quad8(FunctionName,a,b,t0,tc,p1,p2,…)
```

其中 *FunctionName* 要么是用单引号引用的函数文件名(没有后缀“.m”)，要么是由 inline 创建的不带引号的函数变量名。a = a，b = b，t0 = t_0(省略时采用默认值)，p1 等，是参数 p_j，当 $tc \neq [\,]$ 时，*quad8* 提供中间输出。

函数接口形式为：

```
function z = FunctionName(x,p1,p2,…)
```

其中 x 是 *quad8* 的积分变量，它必须总是出现在这一位置上。下面将说明 quad8 的使用方法。

在机械学中经常令人感兴趣的两个量是二维图形的面积和它的重心位置。假定两条曲线 $y_j = f_j(x)$，$j = 1,2$，其交点为 x_1 和 x_2。在曲线两个交点之间的面积可用下式表示：

$$A = \int dA = \int_{x_1}^{x_2} (y_2 - y_1)dx$$

面积相对于原点的重心坐标分别为：

$$x_c = \frac{1}{A}\int x dA = \frac{1}{A}\int_{x_1}^{x_2} x(y_2 - y_1)dx$$

---

① 函数 $g(x,y)$ 的双积分运算用 dblquad 实现，没有参数传递。

$$y_c = \frac{1}{A}\int y\mathrm{d}A = \frac{1}{A}\int_{x_1}^{x_2} \frac{1}{2}(y_2 + y_1)\mathrm{d}A = \frac{1}{2A}\int_{x_1}^{x_2} (y_2^2 - y_1^2)\mathrm{d}x$$

假定 $y_2 = x + 2$ 和 $y_1 = x^2$，如图 5.2 所示。图中直观显示了交点为 $x_1 = -1$ 和 $x_2 = 2$。积分结果产生出 $A = 4.5, x_c = 0.5, y_c = 1.6$。再用数值方法重复这些计算。

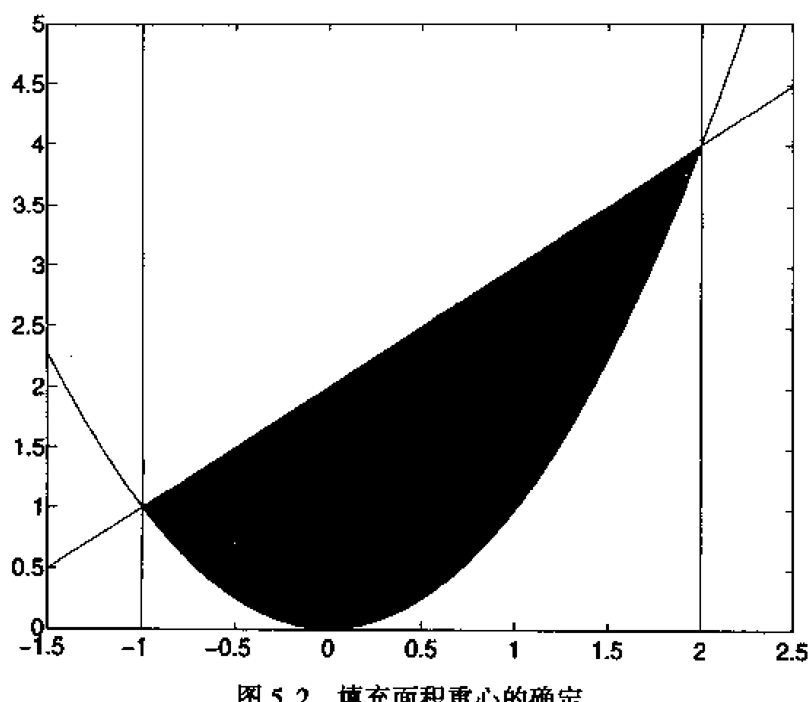


图 5.2　填充面积重心的确定

因为这些曲线的表达式相对简单，所以用 inline 和 quad8 得到如下程序：

```
area = quad8(inline('x+2','x'), -1,2) - quad8(inline('x.^2','x'), -1,2)
xc = quad8(inline('x.*((x+2)-x.^2)','x'), -1,2)/area
yc = quad8(inline('((x+2).^2-x.^4)/2','x'), -1,2)/area
```

执行得到：*area* = 4.5000，*xc* = 0.5000，*yc* = 1.6000。

另一个近似求解的方法是：

```
trapz(x,y)
```

此方式下，可以用数组的形式指定 x 的值和对应的 y 值。函数执行如下运算：首先将 x 分成小的区间，利用相应 y 的平均值与 x 区间相乘求矩形面积。再将所有面积加起来，即为积分结果。尽管这种方法精度差，但它不像 quad8 那样，必须要求创建函数。所以，当积分运算的输入参数仅以数组的形式给出时，这是惟一的办法。现在用 trapz 求解前一个例子：

```
x = linspace(-1,2,150);
y = x + 2 - x.^2;
area = trapz(x,y)
xc = trapz(x,x.*y)/area
yc = trapz(x,(x+2).^2 - x.^4)/2/area
```

程序执行后，用 5 位数字表示的结果是：*area* = 4.4998，*xc* = 0.5000，*yc* = 1.6000。

闭合曲线的面积也可以用下式近似求解：

```
polyarea(x,y)
```

其中(x,y)是对曲线线性化后的近似坐标。在前例中，曲线之一是直线，终点坐标为$(-1,1)$和$(2,4)$。用polyarea计算面积的程序如下：

```
x=linspace(-1,2);
y=x.^2;
x=[-1 2 x];
y=[1 4 y];
area=polyarea(x,y)
```

执行结果为：*area*=4.4995。

空间中一线段的长度可以由下列积分公式求得：

$$L=\int_a^b\sqrt{\left(\frac{dx}{dt}\right)^2+\left(\frac{dy}{dt}\right)^2+\left(\frac{dz}{dt}\right)^2}dt\approx\sum_{i=1}^{N}\sqrt{(\Delta_i x)^2+(\Delta_i y)^2+(\Delta_i z)^2}$$

其中，

$$\Delta_i x=x(t_{i+1})-x(t_i)$$
$$\Delta_i y=y(t_{i+1})-y(t_i)$$
$$\Delta_i z=z(t_{i+1})-z(t_i)$$

并且，$t_1=a,t_{N+1}=b$，参量$\Delta_i x$、$\Delta_i y$和$\Delta_i z$可用下面的函数进行估算：

```
diff
```

它可从向量$\boldsymbol{x}=[x_1x_2\cdots x_n]$求出向量$\boldsymbol{q}$，且$\boldsymbol{q}$有$(n-1)$个元素，见下式：

$$\boldsymbol{q}=[x_2-x_1,x_3-x_2,\cdots,x_n-x_{n-1}]$$

对一向量$\boldsymbol{x}$而言，diff简写为：

```
q=x(2:end)-x(1:end-1);
```

其中 end=length(x)是一个MATLAB函数。

为进一步说明L的近似求解过程，令：

$$x=2t$$
$$y=t^2$$
$$z=\ln t$$

t满足$1\leqslant t\leqslant 2$，且$n=25$，程序如下：

```
t=linspace(1,2,25);
L=sum(sqrt(diff(2*t).^2+diff(t.^2).^2+diff(log(t)).^2))
```

执行结果为$L=3.6931$。

5.5.3 求局部极小值的函数——fminbnd

在优化工具箱中，fminbnd函数可以求出$f(x)$在区间$a\leqslant x\leqslant b$内误差小于t_0的极小值，它能将参数p_j传递到定义$f(x)$的函数中去。fminbnd的一般表达式为：

```
fminbnd(FunctionName,a,b,options,p1,p2,…)
```

其中*FunctionName*要么是由单引号引用的函数文件名（没有后缀“.m”），要么是用inline创建的不带引号的函数变量名。a=a，b=b，*options*是一可选择向量，它的参数用optimset设置（参

见 optimset 的帮助文件)。还有 p1 等这些都称为参数 p_j。

该函数接口形式如下：

```
function z = FunctionName(x,p1,p2,…)
```

其中 x 是自变量，通过改变 x 值 fminbnd 不断使 $f(x)$ 极小化。自变量必须总出现在这个位置。现在说明 fminbnd 的用法。

如图 5.3 所示，MATLAB 示例函数 humps 在 $0 \leqslant x \leqslant 1$ 之间的极小值由下面的语句求出：

```
options = optimset('display0','off');
xmin = fminbnd('humps',0,1,options)
```

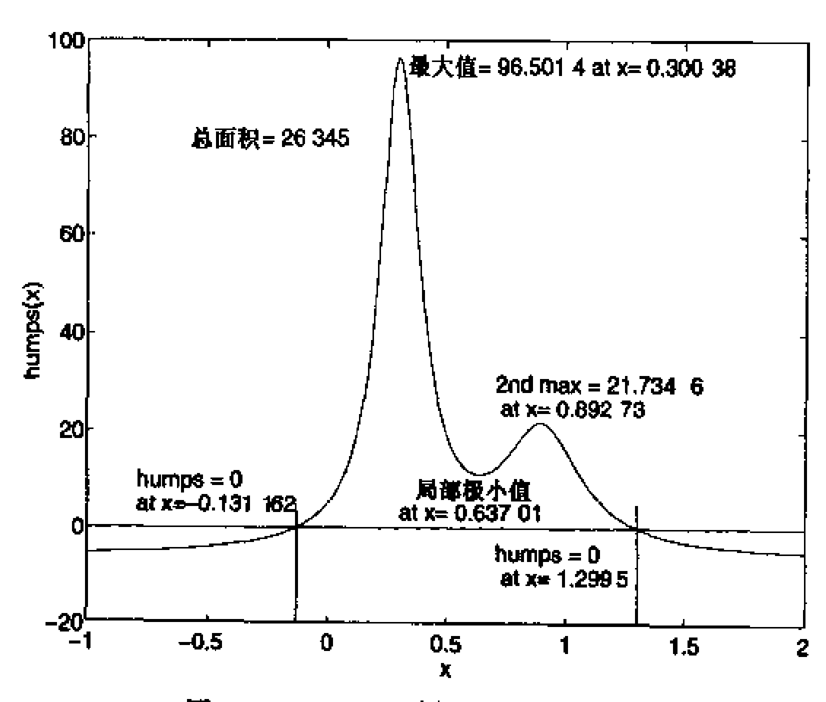


图 5.3 MATLAB 示例函数 humps 的属性

求出 $xmin = x_{min} = 0.6370$。如果想在这一间隔中找到 humps 的极大值，那么用 fminbnd 进行负值操作，或求 humps 的倒数。使用 humps 进行负值操作的程序如下：

```
options = optimset('display0','off');
w = fminbnd(inline('-humps(x)','x'),0,1,options)
```

求出 $w = 0.300376$。humps 在这一点的最大值由下式决定：

```
wmax = humps(0.300376)
```

结果为 $w_{max} = 96.5014$。

利用第 5.5.1 节和第 5.5.2 节讨论的方法可验证图 5.3 中的其他量。

5.5.4 常微分方程的数值求解函数——ode45

函数 ode45 返回 n 个一阶常微分方程的数值解：

$$\frac{dy_j}{dt} = f_j(t, y_1, y_2, \cdots, y_n) \qquad j = 1,2,\cdots,n$$

在 $t_0 \leqslant t \leqslant t_f$ 区间内，初始条件 $y_j(t_0) = a_j, j = 1,2,\cdots,n$，$a_j$ 为常数。ode45 的参数如下：

```
[t,y] = ode45(FunctionName,[t0 tf],[a1 a2 … an]',options,p1,p2,…)
```

其中输出 t 是一时间列向量，由 ode45 确定。输出 y 是数值解的矩阵，行对应时间 t，列对应解的值：

$$y(:,1) = y_1(t)$$
$$y(:,2) = y_2(t)$$
$$\cdots$$
$$y(:,n) = y_n(t)$$

ode45 的第一个参数 *FunctionName* 要么是由单引号引用的函数文件名（不带后缀“.m”），要么是用 inline 创建的不带引号的函数变量名，必须具有如下形式：

```
function yprime = FunctionName(t,y,flag,p1,p2,…)
```

其中 t 是自变量，y 是一向量，它的元素对应于 y_j。*flag* 是一个小写字符串，在本书中未用到它，但必须出现（参见 odefile 帮助文件）。$p1,p2,\cdots$是传递到 *FunctionName* 去的参数，*yprime* 是一个 n 阶列向量，它的元素是 $f_j(t,y_1,y_2,\cdots,y_n),j=1,2,\cdots,n$，记为：

$$yprime = [f_1;f_2;\cdots;f_n]$$

变量名 *yprime*，*FunctionName* 等由程序员指定。

ode45 的第二个参数是二元向量，它给出获得数值解的开始和结束时间。这是一个时间向量$[t_0 t_1 t_2 \cdots t_f]$，在其上得到解。第三个参数是一初始条件向量 $y_j(t_0)=a_j$。第四个参数 options 通常被设定为空（参见 ode45 帮助文件）。其他参数都被传递到函数 *FunctionName*。

在 MATLAB 中还有 4 个常微分方程求解命令，根据微分方程的特点，每个命令各有各的优点。这 4 个命令分别是 ode23、ode113、ode15s 和 ode23s。它们的用法与 ode45 相同。详见 MATLAB 用户指南和各自的帮助文件。

通过求解单变量二阶常微分方程说明函数的用法：

$$\frac{d^2y}{dt^2} + 2\xi\frac{dy}{dt} + y = h(t) \tag{5.1}$$

将上式重写为一个一阶方程组：

$$y_1 = y$$
$$y_2 = \frac{dy}{dt}$$

则为：

$$\frac{dy_1}{dt} = y_2$$
$$\frac{dy_2}{dt} = -2\xi y_2 - y_1 + h$$

考虑三种情况，每一种情况都设 $\xi=0.15,t_0=0,t_f=35$：

情况(1)：

$$y_1(0) = 0$$
$$y_2(0) = 0$$
$$h(t) = u(t)$$

其中 $u(t)$为单位阶跃函数。

情况(2):

$$y_1(0) = 1$$
$$y_2(0) = 0$$
$$h(t) = 0$$

情况(3):

$$y_1(0) = 0$$
$$y_2(0) = 0$$
$$h(t) = \sin(\pi t/5) \quad t \leqslant 5$$
$$= 0 \quad t > 5$$

创建一个函数来处理这三种情况,并以 ode45 所要求的形式提供输出:

```
function s = ForcingFunction(t,w,flag,x,c)
switch c
 case 1
  s = [w(2); - 2 * x * w(2) - w(1) + 1];
 case 2
  s = [w(2); - 2 * x * w(2) - w(1)];
 case 3
  h = sin(pi * t/5). *  (t< = 5)
  s = [w(2); - 2 * x * w(2) - w(1) + h];
end
```

其中 $w(1) = y_1(t), w(2) = y_2(t), x = \xi$。

第 1 种情况:$y_1(0) = 0, y_2(0) = 0, h(t) = u(t)$。程序如下:

```
[tt,yy] = ode45('ForcingFunction',[0 35],[0 0]',[],0.15,1);
plot(tt,yy(:,1))
```

计算结果如图 5.4 所示,为:

$$yy(:,1) = y_1(t) = y(t)$$
$$yy(:,2) = y_2(t) = dy/dt$$

另一种得到常微分方程阶跃响应的方法是使用控制工具箱,参见第 9 章和第 10 章。

第 2 种情况:$y_1(0) = 1, y_2(0) = 0, h(t) = 0$。程序为:

```
[tt,yy] = ode45('ForcingFunction',[0 35],[1 0]',[],0.15,2);
plot(tt,yy(:,1))
```

结果如图 5.5 所示。

第 3 种情况:$y_1(0) = 0, y_2(0) = 0, t \leqslant 5$ 时 $h(t) = \sin(\pi t/5), t > 5$ 时 $h(t) = 0$。 在这种情况下,指定 t 的值,把第二个参数用一时间向量代替。程序为:

```
[tt,yy] = ode45('ForcingFunction',linspace(0,35,200),[0 0]',[],0.15,3);
plot(tt,yy(:,1))
```

结果如图5.6所示。

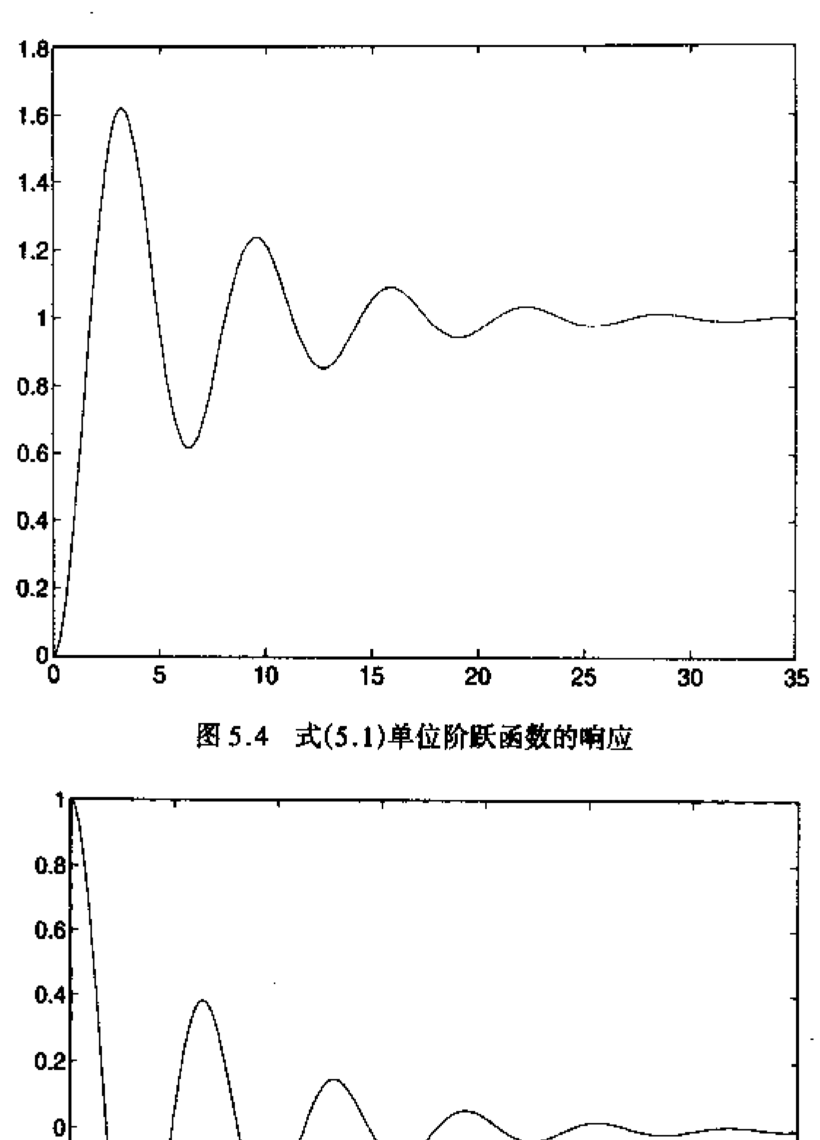


图5.4 式(5.1)单位阶跃函数的响应

图5.5 在初始条件下式(5.1)的响应

举例说明 ode45 的扩展应用。

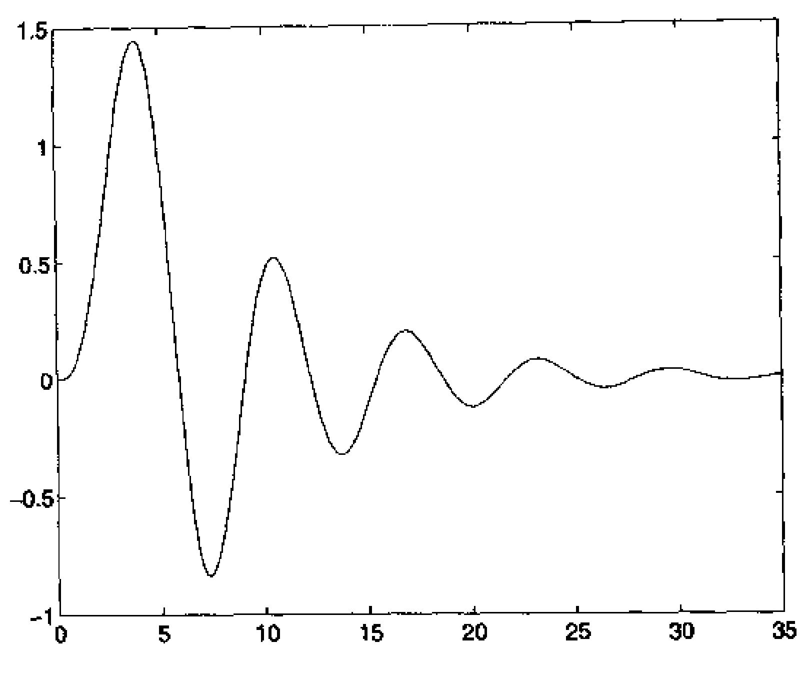


图 5.6 式(5.1)对半正弦波的响应

例 5.1 竖直加热板的自然对流

下式用于说明当和冷液体接触时，竖直加热板的自然对流(参见第 12.3.2 节)：

$$\frac{d^3 f}{d\eta^3} + 3f\frac{d^2 f}{d\eta^2} - 2\left(\frac{df}{d\eta}\right)^2 + T^* = 0$$

$$\frac{d^{2^*} T^*}{d\eta^2} + 3\mathrm{Pr}f\frac{dT^*}{d\eta} = 0$$

当 $P_r = 0.7, \eta = 0$ 时边界条件为：

$$f = 0 \qquad \frac{df}{d\eta} = 0 \qquad \frac{d^2 f}{d\eta^2} = 0.68$$

$$T^* = 1 \qquad \frac{dT^*}{d\eta} = -0.50$$

通过引入中间变量，系统可以分解为五个一阶方程构成的方程组：

$$y_1 = f \qquad y_4 = T^*$$

$$y_2 = \frac{df}{d\eta} \qquad y_5 = \frac{dT^*}{d\eta}$$

$$y_3 = \frac{d^2 f}{d\eta^2}$$

其中 y_1是流函数，y_2是速度，y_3是交量，y_4是温度，y_5是热流量。那么有关这些新变量的微分方程是：

$$\frac{dy_1}{d\eta} = y_2 \qquad \frac{dy_4}{d\eta} = y_5$$

$$\frac{dy_2}{d\eta} = y_3 \qquad \frac{dy_5}{d\eta} = -3\mathrm{Pr}y_1 y_5$$

$$\frac{dy_3}{d\eta} = 2y_2^2 - 3y_1 y_3 - y_4$$

$\eta=0$ 处相应的边界条件为：

$$y_1(0)=0 \qquad y_4(0)=1$$
$$y_2(0)=0 \qquad y_5(0)=-0.50$$
$$y_3(0)=0.68$$

为解这一方程组，首先创建如下函数，指定代表五个一阶微分方程的等式右边的列向量，即 f_j：

```
function ff = NaturalConv(x,y,flag,Pr)
ff = [y(2);y(3); -3 * y(1) * y(3) + 2 * y(2)^2 - y(4);y(5); -3 * Pr * y(1) * y(5)];
```

程序为：

```
y0 = [0 0 0.68 1 -0.50];
Pr = 0.7;
[eta ff] = ode45('NaturalConv',[0 20],y0,[],Pr);
```

结果如图 12.14 所示。

例 5.2 倒置摆

连到一圆盘的倒置摆，如图 10.35 所示，它的线性化运动方程是：

$$ml^2\frac{d^2\theta}{dt^2}+mrl\frac{d^2\Psi}{dt^2}=mgl\theta+b_1\frac{d\theta}{dt}$$
$$mrl\frac{d^2\theta}{dt^2}+(J+mr^2)\frac{d^2\Psi}{dt^2}=b_2\frac{d\Psi}{dt}+\tau_m$$

其中 m 是摆锤的质量，l 是摆的长度，r 是圆盘半径，d 是圆盘的厚度，$J=\rho\pi dr^4/4$ 是圆盘的惯量，b_1是钟摆连接点的摩擦系数，b_2是圆盘的摩擦系数，τ_m是电动机对圆盘底座的转矩。

如果定义下面几个自变量：

$$x_1(t)=\theta(t) \qquad x_3(t)=\frac{d\theta}{dt}$$
$$x_2(t)=\Psi(t) \qquad x_4(t)=\frac{d\Psi}{dt}$$

则以矩阵形式表示的控制方程为：

$$M\dot{x}=Qx+W$$

其中，

$$M=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&ml^2&mlr\\0&0&mlr&J+mr^2\end{bmatrix} \qquad Q=\begin{bmatrix}0&0&1&0\\0&0&0&1\\mgl&0&b_1&0\\0&0&0&b_2\end{bmatrix}$$

$$W=\begin{bmatrix}0\\0\\0\\\tau_m\end{bmatrix} \qquad x=\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix} \qquad \dot{x}=\begin{bmatrix}dx_1/dt\\dx_2/dt\\dx_3/dt\\dx_4/dt\end{bmatrix}$$

为获得 ode45 所要求的四个一阶常微分方程，必须解出方程组的 dx_i/dt。因而，

$$\dot{x} = M^{-1}Qx + M^{-1}W$$

M 的行列式不等于 0,即 $\det(M) = ml^2 J \neq 0$。

首先创建下列函数以指定四个一阶常微分方程等式右边的内容。

```
function p = InvPend(t,x,flag,taum,m,r,L,d,g,rho,b1,b2)
J = 0.25 * pi * rho * d * r^4;
M = [1 0 0 0;0 1 0 0;0 0 m * L^2 m * r * L;0 0 m * r * L J + m * r^2];
Q = [0 0 1 0;0 0 0 1;m * g * L 0 b1 0;0 0 0 b2];
W = [0;0;0;taum];
p = inv(M) * Q * x + inv(M) * W;
```

其中 p 是一列向量。通常可以用 *InvPend* 找到突然施加转矩的系统响应或一系列初始条件下的系统响应。但是,这一系统是不稳定的,形式解不能给出任何有用信息。此外,解会很快离开线性化有效范围。不过,可通过用第 10.5.3 节详细讨论的合适的控制系统使系统稳定。

例 5.3　区域每一端的边界条件

考虑常微分方程:

$$\frac{d^2y}{dx^2} + ky = x \qquad 0 \leqslant x \leqslant 1$$

边界条件为:

$$y(0) = 0 \qquad y(1) = 0$$

因为 ode45 要求仅在 $x = 0$ 处指定初始边界条件,所以必须用一过程决定 $dy(0)/dx$ 以使 $y(1) \to 0$。此过程不断调整 $dy(0)/dx$ 值,使 $y(1)$ 近似为 0[①]。可用 fzero 找到 $dy(0)/dx$ 的值。对于更高阶的微分方程,用 fsolve 而不用 fzero,见第 8.2.1 节和第 12.3.1 节。

首先,用前面提到的方法把方程转换成两个一阶微分方程,如下所示:

$$\frac{dy_1}{dx} = y_2$$

$$\frac{dy_2}{dx} = x - ky_1$$

以 ode45 所要求的形式表示上式的函数是:

```
function f = ExampleODE(x,y,flag,k)
f = [y(2);x - k * y(1)];
```

其次,创建用 fzero 确定 $dy(0)/dx$ 的函数:

```
function z = Slope(s,k)
[x,y] = ode45('ExampleODE',[0 1],[0 s]',[],k);
z = y(end,1);
```

其中 $y(\text{end},1) = y(1)$, $s = dy(0)/dx$。fzero 将改变 s 以使 $z \to 0$。

当 $k = 2$ 时确定 $y(x)$ 的程序如下:

```
options = optimset('display','off');
```

① 这种将边值问题转换为初值问题的方法称为"打靶"方法。

```
k = 2;guess = -0.1;
bc = fzero('Slope',guess,options,k);
[x,y] = ode45('ExampleODE',[0 1],[0 bc]',[],k)
disp(['Slope at x = 0 is'num2str(bc)])
plot(x,y(:,1))
```

执行后,在 MATLAB 命令窗口显示如下结果:

```
Slope at x = 0 is -0.21586
```

并生成如图 5.7 所示的曲线。程序中 $bc = y(1,2)$。此方法的另一个应用见第 11.3.2 节。

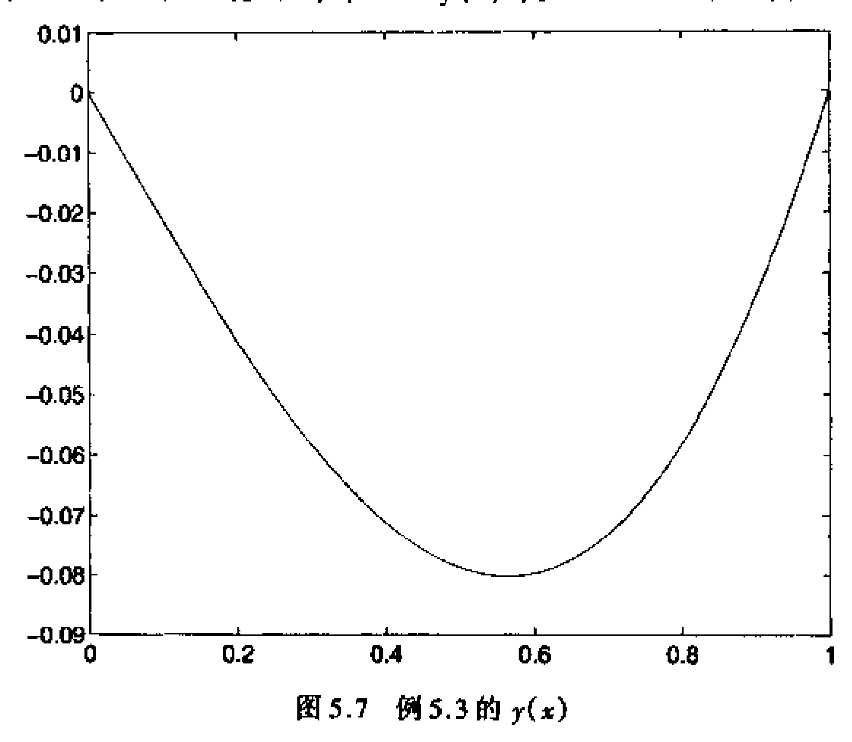


图 5.7 例 5.3 的 $y(x)$

5.5.5 非线性方程的数值求解函数——fsolve

在优化工具箱中的内置函数 fsolve 用于求解非线性方程,如 n 元非线性方程组 $f_n(x_1, x_2, \cdots, x_n) = 0$, x_n 的估计值为 $x_s = [x_{s1}\ x_{s2} \cdots x_{sn}]$。它也可以把参数 p_j 传递到定义 $f_n(x)$ 的函数中。fsolve 的一般表示为:

```
fsolve(FunctionName,xs,options,p1,p2,…)
```

其中 *FunctionName* 要么是由单引号引用的函数名(没有后缀".m"),要么是由 inline 创建的不带引号的函数变量名。*options* 是可选择的向量,它的参数由 *optimset* 设置(参见帮助文件中的 optimset)。$p1, \cdots$ 等统称为参数 p_j。

该函数的接口形式如下:

```
function z = FunctionName(x,p1,p2,...)
z = [f1;f2;...;fn]
```

其中 x 是由 n 个量 x_n 组成的向量,x 和 z 是一列向量,由 x 的 n 个非线性方程 $f_n(x_1, x_2, \cdots, x_n)$ 形式的 MATLAB 表达式组成。

考虑如下方程组,它是图 2.7 所示的三自由度连杆机构的中间求解过程,与倒置摆系统一致。

$$r_1 - a_1\cos(\theta_1) - a_2\cos(\theta_1 + \theta_2) = 0$$

$$r_2 - a_1\sin(\theta_1) - a_2\sin(\theta_1 + \theta_2) = 0$$

为求解这一方程组，首先创建一函数 *kinematics*，它将这些等式变为 fsolve 所要求的形式，如下所示：

```
function w = kinematics(theta,a1,a2,r1,r2)
w = [a1 * cos(theta(1)) + a2 * cos(theta(1) + theta(2)) - r1;...
    a1 * sin(theta(1)) + a2 * sin(theta(1) + theta(2)) - r2];
```

其中 $theta(1) = \theta_1$，$theta(2) = \theta_2$。

令 $r_1 = 1.8$，$r_2 = 2.1$，$a_1 = 1.0$，$a_2 = 2$ 且 θ_1和 θ_2初值设为 $\pi/6$，则程序如下：

```
options = optimset('display','off');
z = fsolve('kinematics',[pi/6 pi/6],options,1,2,1.8,2.1) * 180/pi
```

执行后给出 $\theta_1 = z(1) = 16.6026°$，$\theta_2 = z(2) = 48.5095°$。当 θ_1，θ_2的初值设为 π 时，可求出另一组值。因而使用 fsolve 时必须要注意，特别是存在多个解时。

另一个多解的例子是一椭圆与一条抛物线相交的情形。椭圆的表达式如下：

$$g(x,y) = x^2/4 + y^2 - 1$$

抛物线的表达式为：

$$f(x,y) = y - 4x^2 + 3$$

这两个函数的图形存在四个交点，因而 fsolve 的返回值与初始假设值有关系。通过创建如下函数说明这一点：

```
function w = fgsolve(xy)
w = [0.25 * xy(1).^2 + xy(2).^2 - 1;xy(2) - 4 * xy(1).^2 + 3];
```

其中 $xy(1) = x$，$xy(2) = y$。设初始值 $x = 0.5$，$y = -0.5$，求解的程序如下：

```
options = optimset('display','off');
xy = fsolve('fgsolve',[0.5,-0.5])
```

执行后，得到 $x = xy(1) = 0.7188$，$y = xy(2) = -0.9332$。如果选择初值 $x = -0.5$，$y = 0.5$，则得到 $x = xy(1) = -0.9837$，$y = xy(2) = 0.8707$。

5.6　其他 MATLAB 函数示例

MATLAB 中有许多常用的函数，它们被广泛用于以数字方法求解工程问题，本节用实例说明一些函数的应用。

5.6.1　多项式拟合——polyfit/polyval

金属槽的敏感系数 q 可由 Neuber 常数$\sqrt{a}$和槽口半径 r 定义，表达式如下：

$$q = \left(1 + \frac{\sqrt{a}}{\sqrt{r}}\right)^{-1}$$

$\sqrt{a}$的值对于不同的金属是不同的，它是材料极限强度 S_u 的函数。给定某金属，可通过拟合的

方法，利用多项式近似得到$\sqrt{a}$的数值。一旦有了这个多项式，给定 r 和 S_u 的值即可确定 q 的值。

表5.1给出了钢的数据。利用这些数据，首先确定四阶多项式的系数，此多项式把$\sqrt{a}$看做 S_u 的函数，利用它可获得给定 r 和 S_u 条件下的 q 值。

调用下面的函数可实现拟合过程，获得多项式的系数：

```
polyfit
```

用下面的函数进行计算：

```
polyval
```

若多项式的一般形式如下：

$$y(x) = p_1x^n + p_2x^{n-1} + \cdots + p_nx + p_{n+1} \tag{5.2}$$

那么，获得多项式系数的语句是：

```
p=polyfit(x,y,n)
```

其中 n 是多项式的阶次，$\boldsymbol{p} = [p_1\ p_2 \ldots\ p_n\ p_{n+1}]$是长度为$(n+1)$的向量，它代表式(5.2)中多项式的系数。$x$ 和 y 都是长度为 $m \geq n+1$ 的向量，它们是多项式拟合的数据。$\boldsymbol{x}$ 为输入，$\boldsymbol{y}$ 为输出。

求得 $\boldsymbol{p}$ 后，用下式可求式(5.2)：

```
y=polyval(p,x)
```

其中 $\boldsymbol{p}$ 是一长度为$(n+1)$的向量，它由 polyfit 确定，x 是某一点的标量或向量表示，在其上对多项式进行计算。一般情况下，polyval 中的 p 值可以任意选择。在此，仅关心由 polyfit 确定的值。

表5.1 钢的 Neuber 常数

S_U(ksi)	$\sqrt{a}(\sqrt{\text{in}})$	S_U(ksi)	$\sqrt{a}(\sqrt{\text{in}})$
50	0.130	170	0.028
70	0.092	190	0.020
90	0.072	210	0.015
110	0.057	230	0.010
130	0.046	250	0.007
150	0.037		

为了对表5.1中所列数据进行拟合，并求取$\sqrt{a}$相对于 S_u ($50 \leq S_u \leq 250$)和 r($0 < r < 0.2$)的值编写程序。为简化，假定每次输入一组 S_u 和 r。进而，把表5.1中的数据放在名为 *NeuberData* 的函数中。图6.19(b)给出了一系列曲线，表示 q 值如何随数据变化。

该数据函数如下：

```
function nd=NeuberData
nd=[50,.13;70,.092;90,.072;110,.057;130,.046;150,.037;...
    170,.028;190,.020;210,.015;230,.010;250,.007];
```

其中 $nd(:,1) = Su$，$nd(:,2) = r$。程序如下：

```
ncs=NeuberData
p=polyfit(ncs(:,1),ncs(:,2),4);
r=input('Enter notch radius(0<r<0.2)');
```

```
Su = input('Enter ultimate strength of material(50 < Su < 250)');
q = 1/(1 + polyval(p,Su)/sqrt(r));
disp(['Notch sensitivity = 'num2str(q)])
```

执行这一程序可得到如下结果:

```
Enter notch radius(0 < r < 0.2).1
Enter ultimate strength of material(50 < Su < 250) 135
Notch sensitivity = 0.87999
```

其中程序前两行用户输入 0.1 和 135 两个数,边输入边显示。然后,计算 q 的值并显示在第三行。

5.6.2 插值——interp1

首先参考式(5.1)对应的阶跃函数的数值解,如图 5.4 所示。对系统输出 $p_{overshoot}$ 的超调百分比和上升时间 t_r 进行研究。上升时间定义为系统输出从稳态(长时间)响应 y_x 的 10% 上升到 90% 的时间,y_x 是 $\xi > 0$ 且输入为阶跃函数时的稳态响应。这样:

$$t_r = t_h - t_l$$

其中 t_h 和 t_l 分别由下式决定:

$$y(t_h) = 0.9y_x$$
$$y(t_l) = 0.1y_x$$

超调百分比定义如下。如果输出响应的最大值是 y_{max},并且稳态值是 y_x,则超调百分比为:

$$P_{overshoot} = 100(y_{max} - y_x)/y_x$$

因为 $y(t)$ 已经有了数值解,所以既不用 fzero 也不用 fminbnd。打破这一限制的方法在第 5.6.3 节给出。因而,为找到这些量,用:

```
interp1
```

确定 t_h 和 t_l。并且用:

```
max
```

确定 y_{max} 的位置。interp1 的通用表达式如下:

```
V = interp1(u,v,U)
```

其中 v 是 $v(u)$,u 和 v 是同阶次的向量,U 是一标量,或者是一个由 V 所确定的一系列 u 值构成的向量。V 和 U 的阶次相同,因而程序如下:

```
[t,y] = ode45('ForcingFunction',[0 35],[0 0]',[],0.15,1);
[ymax,tmax] = max(y(:,1));
tr = interp1(y(1:tmax),t(1:tmax),[.1 .9]);
povershoot = 100 * (ymax - y(end,1)/y(end,1);
disp(['Percentage overshoot = 'num2str(povershoot,4)'%'])
disp(['Rise time = 'num2str(tr(2) - tr(1),4)'seconds'])
```

执行后,MATLAB 命令窗口显示如下结果:

```
Percentage overshoot = 61.23%
```

```
Rise time = 1.153 seconds
```

程序的第一行在第5.5.4节已经讨论过了。向量 y 有两列,第一列对应于 $y(t)$,第二列对应于 dy/dt。第二行是函数 max 的另一种输出形式,它对应于第5.2节中的形式 #3(参见 max 的帮助文件)。第二个输出 *tmax* 是数组 $y(:,1)$最大元素值的索引。此时,它是数组的第39个元素(*tmax* = 39)。这样,$y(39,1)$ = *ymax* = 1.6207。interp1 函数要求它的第一个参数是单调的。因而,必须用它的最大值(index = *tmax*)结束数组,否则,在点 $y(:,1)$后数组 $y(:,1)$开始减小。

5.6.3 用 spline 实现数据拟合

下面将产生一些按指数衰减的数据,然后用一系列样条拟合这些数据,使用函数:

```
spline
```

确定振荡周期,并用对数计算它的衰减。

spline 的通用表达式为:

```
Y = spline(x,y,X)
```

其中 y 是$y(x)$,x 和y 是同样长度的向量,用于产生函数关系 $y(x)$。X 是一标量或是 $Y = y(X)$需要的向量。

给定衰减系数 $\xi < 1.0$,对如下函数[①] 在时间 τ 上采样,产生数据:

$$f(\tau,\xi) = \frac{e^{-\xi\tau}}{\cos\alpha}\cos\left(\tau\sqrt{1-\xi^2}+\alpha\right) \tag{5.2}$$

其中,

$$\alpha = \arctan\frac{-\xi}{\sqrt{1-\xi^2}}$$

在函数 *DampedSineWave* 中计算表达式,程序如下:

```
function f = DampedSineWave(tau,xi)
alpha = atan( - xi/sqrt(1 - xi^2));
f = exp( - xi * tau). * cos(tau * sqrt(1 - xi^2) + alpha)/cos(alpha);
```

在 $0\leqslant\tau\leqslant 20$ 范围内采样 $f(\tau,\xi)$的10个等间隔点,将 $\xi = 0.1$ 的原始波形与拟合后的结果多项式画在一起进行比较。程序如下:

```
n = 10;xi = 0.1;
tau = linspace(0,20,n);
data = DampedSineWave(tau,xi);
newdata = linspace(0,20,200);
plot(newdata,spline(tau,data,newdata),'k--',newdata,...
        DampedSineWave(newdata,xi),'k-')
```

程序执行后产生图5.8。可以看出结果相当理想。当等间隔采样点数增加到15时,两条曲线变得几乎不能区分。然而,在下一阶段检查这些拟合数据时,必须让 $n = 40$,以使估计值和实际值相一致。对 plot 的详细讨论见第6.2节。

---

① 参见 S. S. Rao, *Mechanical Vibrations*, Addison-Wesley, Reading, MA, 1986年,第80~82页。

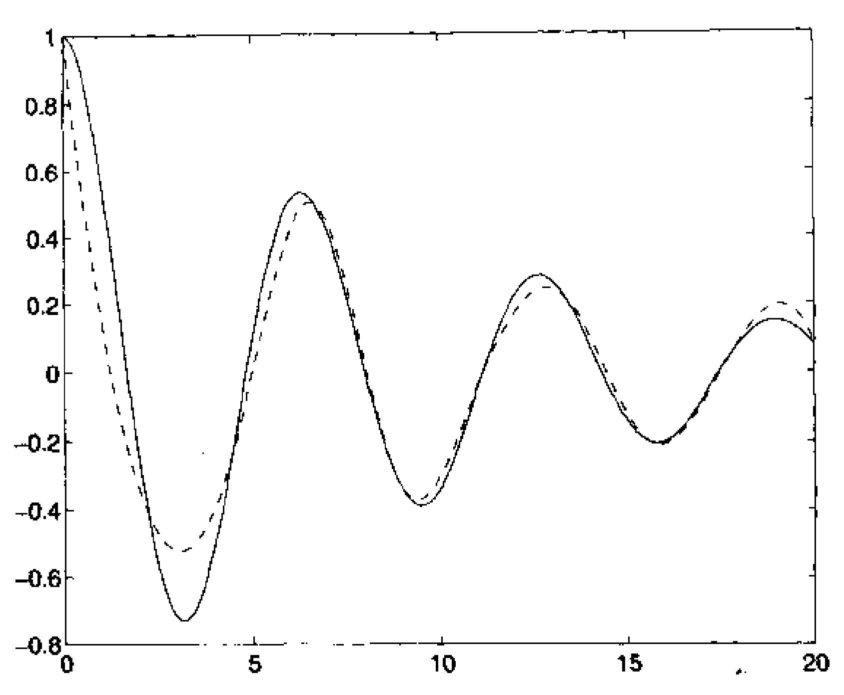


图 5.8 衰减正弦波(实线)与在 $0\leqslant\tau\leqslant20$ 范围内 10 等间隔点采样所获得的近似值波形(虚线)的比较

对数衰减量表示自由衰减振荡幅度的衰减程度①,由下式给出:

$$\Delta = \ln\frac{x_1}{x_2} = \frac{2\pi\xi}{\sqrt{1-\xi^2}}$$

其中,

$$x_1 = f(\tau_1,\xi)$$
$$x_2 = f(\tau_1 + T,\xi)$$

上式由式(5.2)给出,T 为衰减振荡的周期。

由 x_1 和 x_2 确定 ξ 的过程如下。在 $0\leqslant\tau\leqslant20$ 范围内用 spline 拟合 40 个等间隔采样点。然后,创建 200 个时间数据点作为一个向量,利用该向量从拟合函数中计算出衰减正弦波的值。从结果向量中,可以找到极小值点的索引序号。在这些点上,函数 $f(\tau,\xi)$ 近似为极小值。取两倍这样的极小值作为函数 fminbnd 中估计值变化范围的上限,从而确定极小值的精确区间。两倍于最小值时间是周期 T,然后计算 $0.05T$ 和 $1.05T$ 之间幅度变化率的对数值,即值 Δ。再用 fzero 在上式中求解 ξ,实现这些功能的程序如下:

```
n = 40;xi = 0.1;
tau = linspace(0,20,n);
data = DampedSineWave(tau,xi);
tx = linspace(0,20,200);
datafit = inline('spline(tau,data,tx)','tx','tau','data');
[datamin imin] = min(datafit(tx,tau,data));
options = optimset('display','off');
periodT = 2 * fminbnd(datafit,0,2 * tx(imin),options,tau,data);
delta = log(datafit(.05 * periodT,tau,data)/datafit(1.05 * periodT,tau,data));
logdec = inline('s/sqrt(1 - s^2) - d/2/pi','s','d');
```

① 参见 S. S. Rao,出处同上。

```
xiEst = fzero(logdec,[0.01.999],options,delta);
dif = 100 * (xiEst - xi)/xi;
disp(['Estimated xi = 'num2str(xiEst)'Exact xi = 'num2str'(xi)...
        'Difference = 'num2str'(dif)'%'])
```

对于 $\xi = 0.1$，在 MATLAB 命令窗口中显示如下结果：

Estimated xi = 0.10006 Exact xi = 0.1 Difference = 0.063776%

对于 $\xi = 0.95$，显示如下结果：

Estimated xi = 0.94383 Exact xi = 0.95 Difference = -0.64952%

针对后一种情况，若设 $n = 100$，则误差减小到 -0.26%。

5.6.4 数字信号处理——fft 和 ifft

离散傅立叶变换 在 $0 \leqslant t \leqslant T$ 内每隔 Δt 时间对实函数 $g(t)$ 进行一次采样，那么实函数 $g(t)$ 的傅立叶变换可由离散傅立叶变换近似表示为：

$$G_n = G(n\Delta f) = \Delta t \sum_{k=0}^{N-1} g_k e^{-j2\pi nk/N} \qquad n = 0,1,\cdots,N-1$$

其中 $g_k = g(k\Delta t)$，$\Delta f = 1/T$，$T = N\Delta t$，N 是采样数，参见图 5.9。一般而言，G_n 是一复数。对 Δt 的限制是：

$$\alpha\Delta t < \frac{1}{f_h}$$

其中 f_h 是 $g(t)$ 中的最高频率分量，并且 $\alpha \geqslant 2$。量 G_n 称为 $g(t)$ 的幅度密度，其单位是“幅度-秒”，或等价为“增益/Hz”。

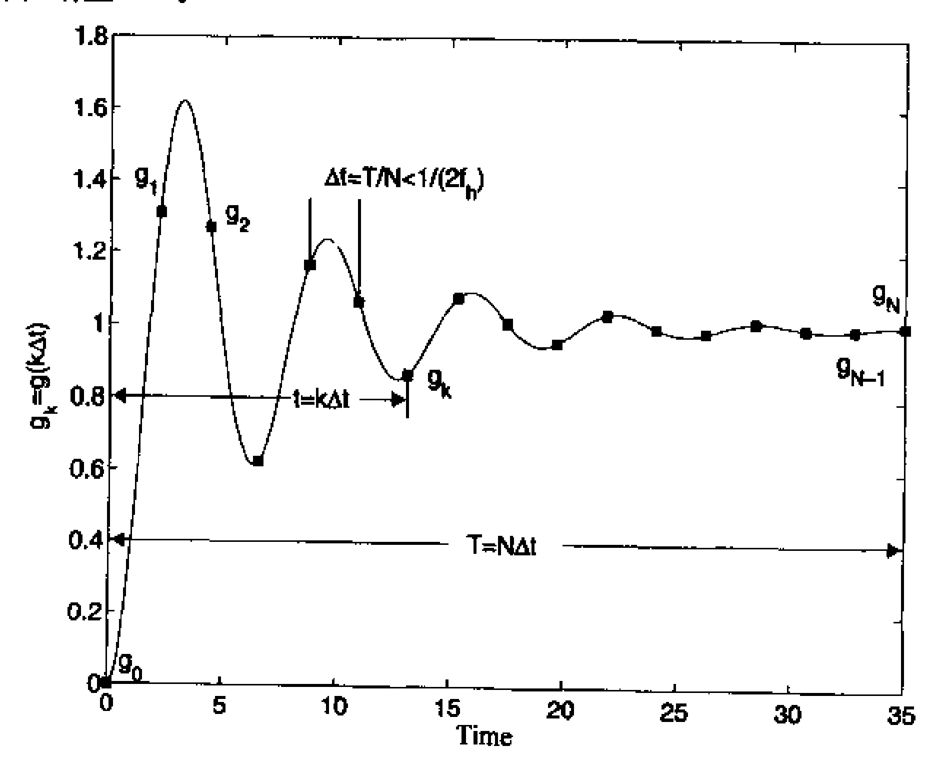


图5.9 采样波形

反变换由下式近似表示：

$$g_k = \Delta f \sum_{n=0}^{N-1} G_n e^{j2\pi nk/N} \qquad k = 0,1,\cdots,N-1$$

为估计出对应于每一个 G_n 在频率 $n\Delta f$ 处的 A_n 值，应计算 G_n 与 Δf 的乘积，即：

$$A_n = \Delta f G_n$$

所以：

$$A_n = \frac{1}{N}\sum_{k=0}^{N-1} g_k e^{-j2\pi nk/N} \qquad n = 0,1,\cdots,N-1$$

因为 $\Delta f \Delta t = 1/N$，所以信号的平均功率为：

$$P_{avg} = \sum_{n=0}^{N-1} |A_n|^2$$

$|A_n|$ 经常作为 $n\Delta f$ 的函数，以得到幅度频谱。此时为：[①]

$$|A_n|_s = 2|A_n| \qquad n = 0,1,\cdots,N/2-1$$

这些表达式用快速傅立叶变换计算最好，是数字计算离散傅立叶变换很有效的算法。当采样数据点数是 2 的整次幂，即 $N = 2^m$ 时较好。其中 m 是正整数。MATLAB 用以下函数实现这一算法：

```
fft(g,N)
```

反变换为：

```
ifft(G,N)
```

其中 fft 返回 $G_n/\Delta t$，ifft 返回 $g_k/\Delta f$。

权函数 在很多种情况下，通过某一适当的函数对 $g(t)$ 加权，以便在另一个变换域中提供更好的解或其他属性。在进行离散傅立叶变换之前，先调整原始的信号，以便消除因窗口函数变化对信号平均值和信号平均功率的影响。如果权函数为 $w_n = w(n\Delta t)$，则被修正信号 g_{cn} 由下式给出[②]：

$$g_{cn} = k_2 w_n (g_n - k_1) \qquad n = 0,1,\cdots,N-1$$

其中，

$$k_1 = \sum_{n=0}^{N-1} w_n g_n \Big/ \sum_{n=0}^{N-1} w_n$$

对窗口函数的平均值修正，并且：

$$k_2 = \left[N \Big/ \sum_{n=0}^{N-1} w_n^2 \right]^{1/2}$$

对窗口函数的平均功率修正。然后对 g_{cn} 进行离散傅立叶变换。

在 MATLAB 的数字信号处理工具箱中包含 8 个常用的权函数。

---

① 参见 J.S.Bendat 和 A.G.Piersol, *Engineering Applications of Correlation and Spectral Analysis*, John Wiley & Sons, New York, 1980 年。

② 参见 E.C.Ifeachor 和 B.W.Jervis, *Digital Signal Processing: A Practical Approach*, Addison-Wesley, Harlow, England, 1993 年，第 593 页。

互相关 两个有限宽的确知信号 $x(t)$和 $y(t)$的互相关由下式给出：

$$R_{xy}(\tau) = \int_{-\infty}^{\infty} x(t)y(t+\tau)\mathrm{d}t \qquad -\infty < \tau < \infty$$

对互密度谱函数 $S_{xy}(\omega)$进行傅立叶反变换可估计出这一量：

$$R_{xy}(\tau) = F^{-1}[S_{xy}(\omega)]$$

其中 $F^{-1}(\cdots)$表示傅立叶反变换，并且：

$$S_{xy}(\omega) = X(\omega)Y^{*}(\omega)$$

$X(\omega)$和 $Y(\omega)$分别是 $x(t)$和 $y(t)$的傅立叶变换。星号表示复共轭。为把 $R_{xy}(\tau)$变换为正确的单位，要求 $S_{xy}(\omega)$乘以 $\Delta t = T/N$。

用两个例子说明这些联系。

例5.4 正弦波的傅立叶变换

令：

$$g(t) = A_o\sin(2\pi f_o t) \qquad 0 \leqslant t \leqslant T = 2^K/f_o \qquad K = 0,1,2,\cdots$$

而且，

$$\Delta t < 1/(2f_o) \text{ 或 } m - K > 1$$

因为：

$$f_h = f_o = 2^K/T$$

且，

$$\Delta t = 2^{-m}T$$

对 $g(t)$用下面的汉明(Hamming)函数加权：

$$\begin{aligned} w(t) &= 0.54 - 0.46\cos(2\pi t/T) \qquad 0 \leqslant t \leqslant T \\ &= 0 \qquad \text{其他} \end{aligned}$$

下面的程序具有的功能是计算和绘制校正后的信号 $g_c(t)$，以及它的幅度谱 A_n，显示信号的平均功率。在此，$P_{avg} = A_o^2/2$。设 $A_o = 2.5, f_o = 10$ Hz, $K = 5$, $m = 10(N = 1024)$。

```
k = 5;m = 10;fo = 10;Ao = 2.5;
N = 2^m;T = 2^k/fo;
ts = (0:N - 1)* T/N;
df = (0:N/2 - 1)/T;
whamm = 0.54 - 0.46 * cos(2 * pi * ts/T);
SampledSignal = Ao * sin(2 * pi * fo * ts);
k1 = sun(whamm. * SampledSignal)/sum(whamm);
k2 = sqrt(N/sum(whamm.^2));
CorrectedSignal = whamm. * (SampledSignal - k1) * k2;
figure(1)
plot(ts,CorrectedSignal)
figure(2)
An = abs(fft(CorrectedSignal,N))/N;
plot(df,2 * An(1:N/2))
disp(['Average power = 'num2str(sum(An.^2))])
```

执行程序得到图5.10和图5.11，并且在MATLAB命令窗口中得到如下信息：

Average power = 3.125

MATLAB 函数 figure 用于提供两个独立的窗口外形，见第 6.1 节。注意，正弦波幅值不等于2.5。当权函数被除去时，可得到真正的幅值。

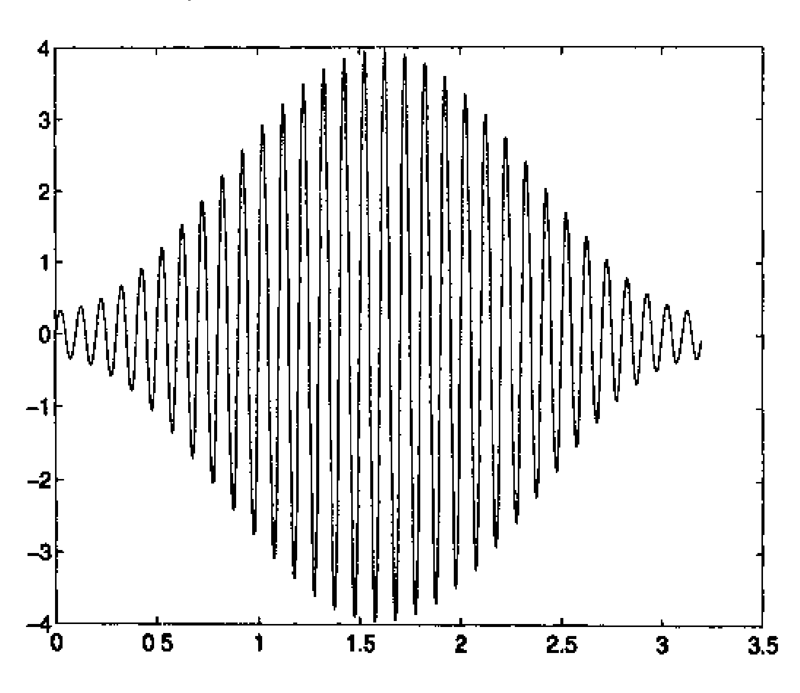


图 5.10　汉明权函数修正的正弦波

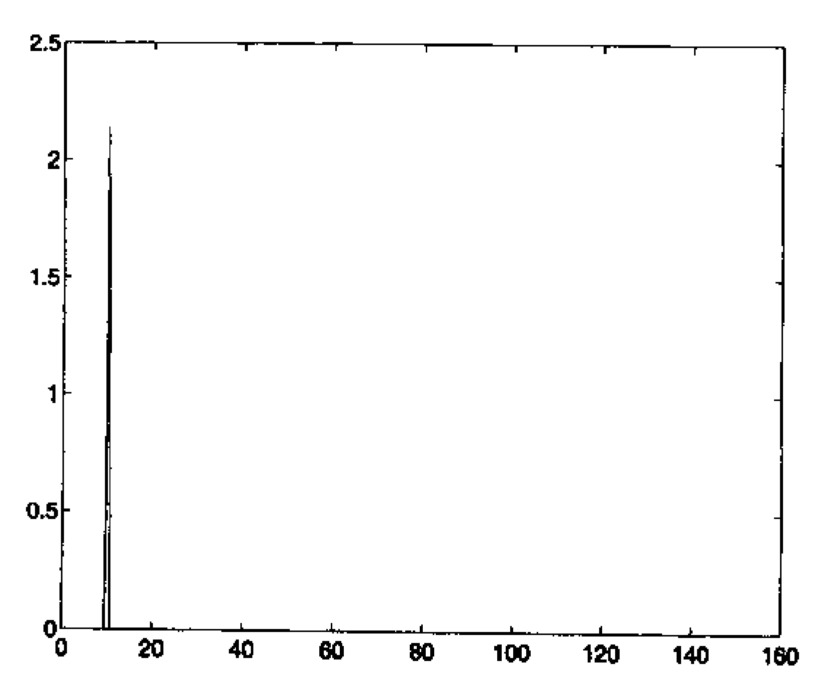


图 5.11　应用汉明权函数的正弦波幅度谱

例 5.5　两脉冲的互相关

对于图 5.12 所示的两个脉冲，互相关函数的表达式如下：

$$x(t) = A_x[u(t) - u(t - T_o)] \qquad t \geqslant 0$$

$$y(t) = A_y[u(t - T_1) - u(t - T_1 - T_2)] \qquad t \geqslant 0$$

其中 $u(t)$为单位阶跃函数。设 $A_x = A_y = 1$, $T_o = 0.01$ s, $T_1 = 2T_o$, $T_2 = T_1 + T_o$, $T = T_2 +$

T_o，$N=2^{10}$。写出如下程序：

```
To = 0.01;T1 = 2 * To;T2 = T1 + To;Tend = T2 + To;
N = 2^10;deltaT = Tend/N;ampl = 1;
t = linspace(0,Tend,N);
PulseCroseeCorr = inline('ampl * ((t - Ts > = 0) - (t - Te > 0))','t','Ts','Te','ampl');
x = PulseCrossCorr(t,0,To,ampl);
y = PulseCrossCorr(t,T1,T2,ampl);
X = fft(x,N);
Y = conj(fft(y,N));
Rxy = ifft(X. * Y * deltaT,N);
plot(t,real(Rxy))
```

程序的执行结果见图 5.13。由于数值计算会带来截断误差，所以函数 real 用来除去残留的虚部。

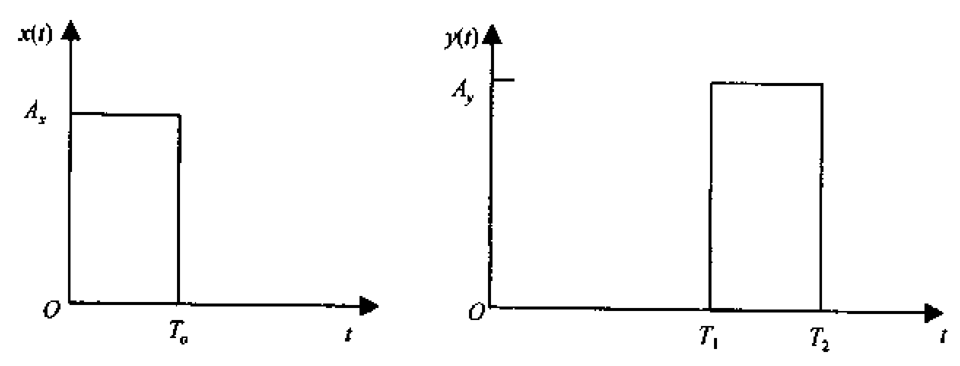


图 5.12　两　脉　冲

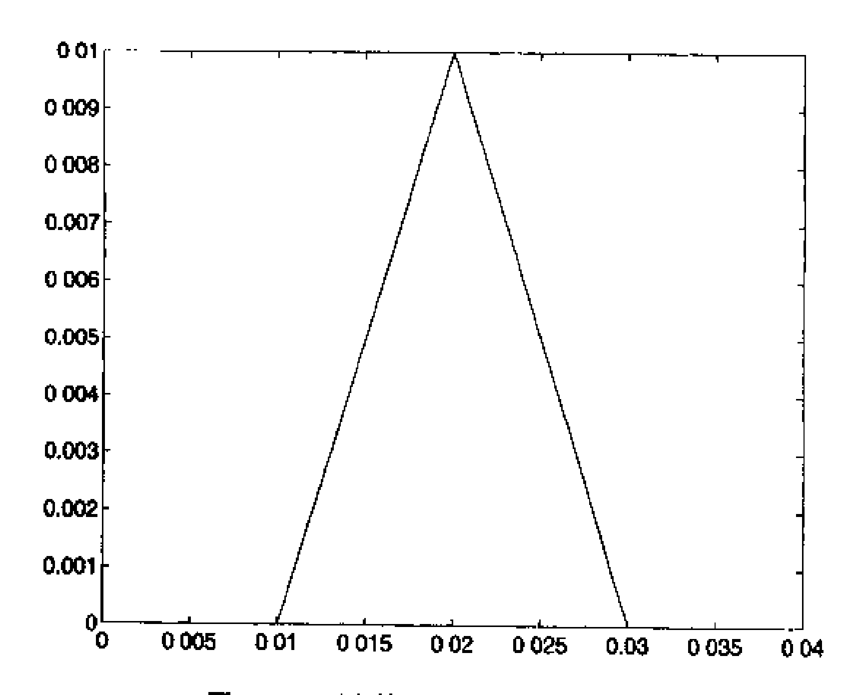


图 5.13　两个等周期脉冲的互相关函数

练习

关于第 5.5.1 节内容的练习：

5.1 由下面多项式的根确定主应力:[①]

$$\sigma^3 - C_2\sigma^2 - C_1\sigma - C_0 = 0$$

其中,

$$C_2 = \sigma_x + \sigma_y + \sigma_z$$
$$C_1 = \tau_{xy}^2 + \tau_{yz}^2 + \tau_{zx}^2 - \sigma_x\sigma_y - \sigma_y\sigma_z - \sigma_z\sigma_x$$
$$C_0 = \sigma_x\sigma_y\sigma_z + 2\tau_{xy}\tau_{yz}\tau_{zx} - \sigma_x\tau_{yz}^2 - \sigma_y\tau_{zx}^2 - \sigma_z\tau_{xy}^2$$

$\sigma_x,\sigma_y,\sigma_z$为施加的法向应力;$\tau_{xy},\tau_{yz},\tau_{zx}$为施加的切向应力。如果方程的根是三个主应力 σ_1、σ_2和 σ_3,且 $\sigma_1 > \sigma_2 > \sigma_3$,则切向主应力为:

$$\tau_{12} = (\sigma_1 - \sigma_2)/2 \qquad \tau_{23} = (\sigma_2 - \sigma_3)/2 \qquad \tau_{13} = (\sigma_1 - \sigma_3)/2$$

其中 $\tau_{max} = \tau_{13}$。

确定主应力值,即可得到相应的切向应力值:

$$\sigma_x = 100 \qquad \tau_{xy} = -40$$
$$\sigma_y = -60 \qquad \tau_{yz} = 50$$
$$\sigma_z = 80 \qquad \tau_{zx} = 70$$

寻根函数未对根进行排序。可使用 sort 函数按照第2.5.4节例2.2所讨论的方式对根进行排序。

[答案:$\sigma_1 = 160.7444, \sigma_2 = 54.8980, \sigma_3 = -95.6424, \tau_{12} = 52.9232, \tau_{23} = 75.2702, \tau_{13} = 128.1934$。]

5.2 用 fzero 函数求取下列方程最接近的五个正根,特殊声明除外。函数仅要求在[x_0 x_1]区域内进行搜索。确定根以前绘制函数图形,可使用 axis 提高图形分辨率。

(a)弹簧的振动方程:[②]

$$\tan x = x$$

(b)平板的对流换热方程。[③] $p = 0.1$ 和 $p = 1$ 时分别求取方程的根。

$$2\cot x = \frac{x}{p} - \frac{p}{x}$$

(c)环形薄膜的振动方程,[④] 设 $b = 2$。

$$J_0(x)Y_0(xb) - J_0(xb)Y_0(x) = 0$$

用 besselj 和 bessely 分别求取一类零阶贝塞尔函数 $J_0(x)$和二类零阶贝塞尔函数 $Y_0(x)$。

(d)一根悬臂梁,自由端附加集中质量 M_0,其振动方程[⑤]如下所示。当 $M_0/m_0 = 0$, 0.2 和 1 时分别求根。

$$(M_0/m_0)\Omega[\cos(\Omega)\sinh(\Omega) - \sin(\Omega)\cosh(\Omega)] + \cos(\Omega)\cosh(\Omega) + 1 = 0$$

---

① 参见 J E.Shigley 和 C.R.Mischke, *Mechanical Engineering Design*,第五版,McGraw-Hill,New York,1998 年。

② 参见 E.B.Magrab, *Vibration of Elastic Structural Members*,Sythoff & Noordhoff,The Netherlands,1979 年,第 58 页。

③ 参见 M N.Ozisik, *Heat Conduction*,第二版,John Wiley & Sons,New York, 1993 年,第 47 页。

④ 参见 E B Magrab,出处同上,第 83 页。

⑤ 参见 E.B Magrab,出处同上,第 130 页

(e)一端夹紧,另一端支撑的简支梁振动方程[1]。

$$\tanh(\Omega) - \tan(\Omega) = 0$$

(f)边沿夹紧的固体圆盘的振动方程[2] 如下所示:

$$J_m(\Omega)I_{m+1}(\Omega) + I_m(\Omega)J_{m+1}(\Omega) = 0$$

其中 $J_m(x)$是一类 m 阶贝塞尔函数,$I_m(x)$是一类 m 阶贝塞尔修正函数。分别用 besselj 和 besseli 求 $J_m(x)$和 $I_m(x)$。求取 $m = 0$、1 和 2 时最接近的三个根。保存程序用于练习 7.4。

(g)下面的方程[3] 来源于悬索对称模式的求取。求解当 $\lambda^2 = 2\pi^2, 4\pi^2, 8\pi^2$时的最接近根。这一解必须通过交替画图得到。用 axis 函数将纵轴限制在从 -10 到 20 的范围内,见第 6.2 节。

$$\tan\Omega = \Omega - \frac{4\Omega^3}{\lambda^2}$$

(h)分析梯形截面开口管道内不均匀流动时,流体深度与能量梯度 x 的高度比由下式确定。[4]

$$(1 + c_0x)^2(x^2 - x^3) = c_1$$

其中$0 \leqslant c_0 \leqslant 11$ 和 $0.005 \leqslant c_1 \leqslant 12.3$ 是管道的几何形状和流速的函数。然而,并不是所有 c_0和 c_1的组合都是合适的。在 0 和 1 之间求一对实数值 x,并在:(1) $c_0 = 0.4, c_1 = 0.2$;(2) $c_0 = 7.0, c_1 = 4.0$ 条件下使方程成立。用 fzero 和 roots 两种方法求解方程。用 roots 求解时方程被重写为:

$$-c_0^2x^5 + (c_0^2 - 2c_0)x^4 + (2c_0 - 1)x^3 + x^2 - c_1 = 0$$

(i)在开口管道流体顶部的扰动波内,流体的流速大于波动速度,波动角β($0 < \beta \leqslant \pi/2$)由下式确定。[5]

$$2N_F^2\sin^2(\beta)\tan^2(\beta - \theta) = \tan(\beta)\tan(\beta - \theta) + \tan^2(\beta) \qquad \beta > \theta$$

其中 θ 为墙偏转角,$1 \leqslant N_F \leqslant 12$ 为弗劳德(Froude)数。当 $\theta = 35°, N_F = 5$ 时,求 β 值($0 < \beta \leqslant 90°$)。首先画出函数图形。

(j)内部收益率 i_r作为投资结余所获利润的百分比,可由下式[6] 确定。

$$\sum_{k=0}^{n} F_k(1 + i_r)^{-k} = 0$$

其中 n 是期数,i_r是以十进制表示的内部收益率,F_k是每期的现金流量:正的现金流量表示现金吸纳;负的现金流量表示现金支付。当 $F_0 = -\$1000, F_1 = -\$800, F_2 = \$500, F_3 = \$500, F_4 = \$500, F_5 = \$1200$ 时,求 i_r。

(k)如果投资总额为 P,该期内收益额为 A,那么每期利率为 i 时,P 的投资回收期 n

---

① 参见 E.B.Magrab,出处同上,第 130 页。

② 参见 E.B.Magrab,出处同上,第 252 页。

③ 参见 M.Irvine, *Cable Structures*, Dover Publications, Inc., New York, 1981 年,第 95 页。

④ 参见 H.W.King, *Handbook of Hydraulics*,第四版,McGraw-Hill, New York, 1954 年,第 81 页。

⑤ 参见 N.H.C.Hwang 和 C.E.Hita, *Fundamentals of Hydraulic Engineering Systems*,第二版,Prentice Hall, Englewood Cliffs, NJ, 1987 年,第 222 页。

⑥ 参见 G.J.Theusen 和 W.J.Fabrycky, *Engineering Economy*,第八版,Prentice-Hall, Englewood Cliffs, NJ, 1993 年,第 176 页。

由下式确定。[①]

$$\frac{A}{P}=\frac{i(1+i)^n}{(1+i)^n-1}$$

如果 $i=12\%$ /年，$A/P=0.16$，那么确定 n，即收回投资的年数。

(l)威布尔(Weibull)概率密度函数中参数 β 的估计值(见第 14.2.2 节)由下述方程的解确定：[②]

$$\beta=\left[\frac{\sum_{i=1}^{n}x_i^\beta\ln(x_i)}{\sum_{i=1}^{n}x_i^\beta}-\frac{1}{n}\sum_{i=1}^{n}\ln(x_i)\right]^{-1}$$

其中 x_i 是大小为 n 的随机采样。如果：

$x=[72\ 82\ 97\ 103\ 113\ 117\ 126\ 127\ 127\ 139\ 154\ 159\ 199\ 207]$

确定 β 值。

(m)在确定一个球体和平面的表面接触剪切应力时(它是轴承相对于表面的影响模型)，比例 x 的值由下式获得。[③]

$$x\ln(\sqrt{x^2-1}+x)-\sqrt{x^2-1}-Cx=0$$

其中 $x>1$，$C<1$。当 $C=0.5$ 时，确定 x 值。

5.3 求解下列方程的三个实根：[④]

$$x^4=2^x$$

[提示：首先在 $-1\leqslant x\leqslant 2$ 和 $2\leqslant x\leqslant 17$ 两个不同区域中绘制函数图形。]

5.4 某气体的可压缩性系数为 Z，其广义方程的计算公式由下式给出：[⑤]

$$\begin{aligned}Z(r,\tau)=&1+r\sum_{i=1}^{6}A_i\tau^{i-1}+r^2\sum_{i=7}^{10}A_i\tau^{i-7}+r^3\sum_{i=11}^{13}A_i\tau^{i-11}+r^4A_{14}\tau\\&+r^5(A_{15}\tau^2+A_{16}\tau^3)+r^6A_{17}\tau^2+r^7(A_{18}\tau+A_{19}\tau^3)\\&+r^8A_{20}\tau^3+r^2e^{-0.0588r^2}[A_{21}\tau^3+A_{22}\tau^4+r^2(A_{23}\tau^3+A_{24}\tau^5)\\&+r^4(A_{25}\tau^3+A_{26}\tau^4)+r^6(A_{27}\tau^3+A_{28}\tau^5)+r^8(A_{29}\tau^3+A_{30}\tau^4)\\&+r^{10}(A_{31}\tau^3+A_{32}\tau^4+A_{33}\tau^5)]\end{aligned}$$

其中 $\tau=T_c/T(0.4\leqslant\tau\leqslant 1)$；$r=RT_c/P_c\nu$；$R$ 为气体常数，单位为(MPa-m³)/(kg-K)；T 为温度，单位为 K；P 为压强，单位为 MPa；ν 为比体积，单位为 m³/kg；T_c 和 P_c 分别为临界温度和临界压力，33 个常数值由表 5.2 给出。

(a)创建一个函数以确定 $Z(r,\tau)$。用下列检验值检查函数，并使用 `format long e`

---

① 参见 G. J. Theusen 和 W. J. Fabrycky，出处同上，第 188 页。

② 参见 D. C. Montgomery 和 G. C. Runger, *Applied Statistics and Probability for Engineers*, John Wiley & Sons, New York, 1994 年，第 299 页。

③ 参见 W. Changsen, *Analysis of Rolling Element Bearings*, Mechanical Engineering Publishers, London, 1991, 第 80 页。

④ 参见 Problem suggested by Prof. Jeffery M. Cooper, Department of Mathematics, University of Maryland, College Park, MD。

⑤ 参见 W. C. Reynolds, "Thermodynamic Properties in SI," Department of Mechanical Engineering, Stanford University, Stanford, CA, 1979 年。

选项：

(i)Z(1,1) = 0.702 423 969 27

(ii)Z(1/0.3,1) = 0.299 999 999 80

(iii)Z(2.5,0.5) = 0.992 218 539 28

为满足练习 6.12 中的应用，保存该函数以及(b)和(c)中所创建的函数。

(b)上面各量应用于下式：

$$Z(r,\tau) = \frac{p\tau}{r} = \frac{Pv}{RT} \tag{a}$$

其中 $p = P/P_c(1 \leqslant p \leqslant 6)$。使用式(a)在下列条件下确定 r 和 $Z(r,\tau)$的值：(i)$p = 0.6, \tau = 1/1.05$；(ii)$p = 2.18, \tau = 1/1.2$。

[答案：(i)r = 0.7131 时 Z = 0.8013；(ii)r = 3.3567 时 Z = 0.5412。]

(c)使用式(a)在下列条件下确定 τ 和 $Z(r,\tau)$的值：(i)$p = 0.6, r = 1/1.4$；(ii)$p = 2.18, r = 1/0.6$。

[答案：(i)$\tau = 0.9532$ 时 $Z = 0.8007$；(ii)$\tau = 0.6505$ 时 $Z = 0.8508$。]

表 5.2　广义公式中Z的常数

j	A_j	j	A_j	j	A_j
1	0.062 432 384	12	−0.000 727 155 024 313	23	−0.084 519 449 381 3
2	0.127 214 77	13	−0.004 524 546 526 10	24	−0.003 409 313 119 28
3	−0.936 332 33	14	0.001 304 687 241 00	25	−0.001 951 270 499 01
4	0.701 844 11	15	−0.000 222 165 128 409	26	$4.938\,999\,109\,78 \times 10^{-5}$
5	−0.351 608 96	16	−0.001 981 405 356 56	27	$-4.932\,646\,129\,30 \times 10^{-5}$
6	0.056 450 032	17	$5.975\,739\,729\,21 \times 10^{-5}$	28	$8.856\,665\,723\,82 \times 10^{-7}$
7	0.029 956 146 990 7	18	$-3.641\,353\,497\,02 \times 10^{-6}$	29	$5.347\,880\,295\,53 \times 10^{-8}$
8	−0.031 817 436 764 7	19	$8.413\,648\,453\,86 \times 10^{-6}$	30	$-5.934\,205\,591\,92 \times 10^{-8}$
9	−0.016 821 105 551 7	20	$-9.828\,688\,588\,22 \times 10^{-9}$	31	$-9.068\,133\,269\,22 \times 10^{-9}$
10	1.602 040 600 81	21	−1.576 830 568 10	32	$1.618\,224\,072\,65 \times 10^{-9}$
11	−0.001 099 967 407 46	22	0.040 072 898 890 8	33	$-3.320\,447\,939\,15 \times 10^{-10}$

5.5 作为管道摩擦系数 λ 的函数，管道中流动的流体压降可由阔尔布鲁克(Colebrook)公式进行估算：[①]

$$\lambda = \left[-2\log_{10}\left(\frac{2.51}{R_e\sqrt{\lambda}} + \frac{0.27}{d/k}\right)\right]^{-2} \qquad R_e \geqslant 4000$$

其中 R_e是雷诺数，d 是管道直径，k 是表面粗糙度。对于光滑管道($k \approx 0$ 或 $d/k > 100000$)：

$$\lambda = \left[-2\log_{10}\left(\frac{R_e\sqrt{\lambda}}{2.51}\right)\right]^{-2} \qquad R_e \geqslant 4000$$

对充分掺混的紊流流动而言，摩擦系数由下式给定：

$$\lambda = \left[2\log_{10}\left(3.7\frac{d}{k}\right)\right]^{-2}$$

它独立于 R_e。它是阔尔布鲁克通用公式的一个特例，用于获得迭代类型问题的初

① 参见 N H.C.Hwang 和 C.E Hita，出处同上，第 68 页。

值,如第 11.2.3 节所述。

如果 λ 值的范围从 0.008 到 0.08,则当 $R_e = 10^5$,(1) $d/k = 200$,(2) $k = 0$ 时求 λ 值。

保存函数和程序以用于练习 6.11。

[答案:(1)$\lambda = 0.0313$,(2)$\lambda = 0.0180$。]

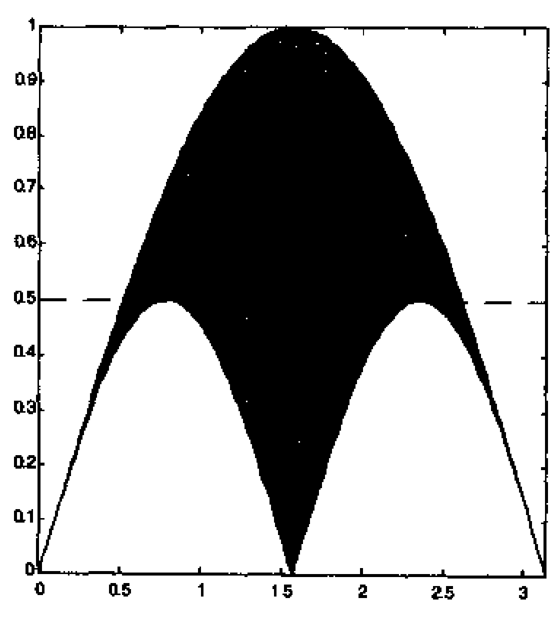


图 5.14 练习 5.6 用图

关于第 5.5.2 节内容的练习:

5.6 用 quad8 和 trapz 求取图 5.14 所示的两个正弦曲线之间的面积。

5.7 为确定偏心负荷下轴向推力轴承的负荷分布,必须要进行如下的积分运算:[①]

$$I_m(\varepsilon) = \frac{1}{2\pi}\int_{-a}^{a}[1 - (1 - \cos(x))/2\varepsilon]^c \cos(mx)\mathrm{d}x$$

其中 $\varepsilon > 0$, $m = 0$ 或 1,且:

$$a = \arccos(1 - 2\varepsilon)$$

滚珠轴承 $c = 1.5$,滚柱轴承 $c = 1.1$。确定滚珠轴承 $I_1(0.6)$的值。

[答案:$I_1(0.6) = 0.2416$。]

5.8 给定:

$$\int_0^\infty E_{\lambda,b}(\lambda, T)\mathrm{d}\lambda = \sigma T^4$$

其中,

$$E_{\lambda,b}(\lambda, T) = \frac{C_1}{\lambda^5[\exp(C_2/\lambda T) - 1]}$$

λ 为波长,单位为 μm;T 为温度,单位为 K;$C_1 = 3.742 \times 10^8 \mathrm{W \cdot \mu m^4/m^2}$;$C_2 = 1.439 \times 10^4 \mathrm{\mu m \cdot K}$;$\sigma = 5.667 \times 10^{-8} \mathrm{W/m^2 \cdot K^4}$为 Stephan-Boltzmann 常数。$T = 300, 400, 500$ K 时

① 参见 W. Changsen,出处同上,第 92 页。

进行数值积分，并计算结果的相对误差。因为两个积分限很难用数字量给出，因而近似积分时采用下限 1 μm 和上限 150 μm。积分限由上述三个温度值处的 $E_{\lambda,b}$图和不引起 quad8 产生警告信息的积分限值确定。

[答案：$\text{error}_{300}=-0.145\%$，$\text{error}_{400}=-0.061\%$，$\text{error}_{500}=-0.030\%$。]（可参见练习 12.6。）

关于第 5.5.3 节内容的练习：

5.9 蜗轮超前角为 λ，比率 $\beta=N_1/N_2$，其中 N_1为蜗轮齿数，N_2为从动轮齿数，轴间中心距为 C，法向径节 P_{dn}为：[1]

$$K=\frac{2P_{dn}C}{N_2}=\frac{\beta}{\sin\lambda}+\frac{1}{\cos\lambda}$$

在实际中，$1\leqslant K\leqslant 2$，$1°\leqslant\lambda\leqslant 40°$，$0.02\leqslant\beta\leqslant 0.30$。对特定值的组合，$\lambda$ 可有一个值、两个值或无值。

(a)当 $\beta=0.02$、0.05、0.08、0.11、0.15、0.18、0.23 和 0.30 时，求使 K 最小的 λ 值。保存程序以用于练习 6.8。

(b)$K=1.5$，$\beta=0.16$ 时，求 λ 值。

5.10 考虑式(5.1)和它在阶跃输入的数值解。确定使下述量最小的 ξ 值：

$$f(\xi)=\sum_{n=1}^{N}(y(t_n)-1)^2$$

令 ξ 在 0.05 到 1.5 的范围内以 0.05 的增量递增。因为 $f(\xi)$是一系列的数字值，所以不能使用 fminbnd。用 min 返回 ξ 的索引，并求取在该索引时的值。

5.11 练习 1.7 中，从压力为 p_0，并处于可逆绝热条件下的容器中溢出气体的质量流量和下式成比例：

$$\Psi=\sqrt{\frac{k}{k-1}}\sqrt{\left(\frac{p_e}{p_0}\right)^{2/k}-\left(\frac{p_e}{p_0}\right)^{(k+1)/k}}$$

其中 p_e是容器的外部压力，k 是绝热可逆气体常数。极大值出现在：

$$\frac{p_e}{p_0}=\left(\frac{2}{k+1}\right)^{\frac{k}{k-1}}$$

$k=1.4$ 时，用 fminbnd 和 min 验证这一极大值，min 利用 $0\leqslant p_e/p_0\leqslant 1$ 范围内的 200 个等间隔值进行计算。

关于 5.5.4 节内容的练习：

5.12 一枚导弹，以初始速度 v_0，水平夹角 α 离开原点(0,0)。如果导弹在(x_e,y_e)点着陆，且在飞行中受到一拉力，其大小和速度的平方成比例，那么控制导弹飞行轨迹的四个一阶方程为：[2]

---

① 参见 M.F.Spotts 和 T.E.Shoup, *Design of Machine Elements*, Prentice Hall, Upper Saddle River, NJ, 1998 年，第 613 页。

② 参见 H.B.Wilson 和 L.H.Turcotte, *Advanced Mathematics and Mechanics Applications Using MATLAB*, 第二版, CRC Press, Boca Raton, FL, 1997 年，第 294 页。

$$\frac{dv_x}{dx}=-c_d v \qquad \frac{dv_y}{dx}=\frac{-(g+c_d v v_y)}{v_x} \qquad \frac{dy}{dx}=\frac{v_y}{v_x} \qquad \frac{dt}{dx}=\frac{1}{v_x}$$

其中 y 是导弹的垂直高度，x 是飞行的水平距离，t 是时间，v_x和 v_y分别是速度 v 的水平和垂直分量。c_d是拉力系数，g 是重力加速度，且：

$$v=\sqrt{v_x^2+v_y^2}$$

这些方程仅当 v_0足够大，以至于当到达 x_e，v_x大于零时才有效。条件测试可以判断$|v_x|>v_0\times10^{-6}$的状态。如果条件不满足，则必须要终止程序的执行，终止操作用 error 实现。条件检查应放置在 ode45 函数调用的开始处。初始条件为：

$$v_{0x}=v_0\cos(\alpha) \qquad v_{0y}=v_0\sin(\alpha) \qquad y=0 \qquad t=0$$

令 $y_1(x)=v_x, y_2(x)=v_y, y_3(x)=y, y_4(x)=t$，依次写出各方程。

(a)当 $v_0=600$ f/s，$c_d=0.002$，$\alpha=45°$时，画出导弹到达 $y_e=0$ 前的飞行轨迹，即在 $y_e>0$ 的条件下绘图。ode45 中令 $x_{final}=1000$ ft。

(b)求导弹的最大上升高度及出现这一高度时的飞行距离。用 fminbnd 和 spline 求解。

[答案：$x=648.1205$ ft 时 $y_{max}=474.8285$ ft。]

(c)求 $y_e=0$ 时的 x_e值及到达这一点时所需的时间。用 interp1 求解。

[答案：$x_e=975.3240$，所需时间为 10.6246 s。]

5.13 一个蹦极爱好者准备从一高空热气球跳下，所用橡皮带长为 L。为保证安全，必须要预知最大加速度、速度和总下落高度，确保使力不会太大而且气球足够高以保证蹦极者不会撞到地面。考虑空气动力学阻力，控制方程为：①

$$\frac{d^2x}{dt^2}+c_o\text{signum}(dx/dt)\left(\frac{dx}{dt}\right)^2+\frac{k}{m_J}(x-L)u(x-L)=g$$

其中 $g=9.8$ m/s²为重力加速度；c_d和阻力系数成比例，单位为 m⁻¹；k 为橡皮带的弹性系数，单位为 N/m；m_J为蹦极者的质量；$u(z)$为单位阶跃函数，即：$z\leqslant0$ 时，$u(z)=0$；$z>0$ 时，$u(z)=1$。如采用第 4.1 节所述的逻辑运算符描述 $u(t)$，则程序将大为简化。

如果 $L=150$ m，$m_J=70$ kg，$k=10$ N/m，$c_0=0.00324$ m⁻¹，初始条件为零，试证明：

(1)11.47 s 时，x 的最大值为 −308.47 m。

(2)5.988 s 时，蹦极者将下降 150 m，速度为 −43.48 m/s。

(3)11.18 s 时，最大加速度将是 −12.82 m/s²(−1.308 g)。

画出位移、速度和加速度曲线。加速度由 diff 求得的速度的近似导数获得。对 ode45 的输出应用 spline 可得到上述数值解。

5.14 一个由长为 L 的无重量刚性棒组成的倒置摆，其自由末端系有一个质量为 m 的物体和一个弹性常数为 k 的线性弹簧。钟摆开始时处于垂直位置，且未拉伸时，弹簧的长度为 L。摆轴的转动阻尼为 c，摆由力矩 $M(t)$驱动。则描述倒置摆角运动的

① 参见 D. M. Etter, *Engineering Problem Solving with MATLAB*, Prentice Hall, Upper Saddle River, NJ, 1997 年，第 220～221 页。

控制方程为：[1]

$$\frac{d^2\theta}{d\tau^2}+\alpha\frac{d\theta}{d\tau}-\sin\theta+\beta\left(1-\frac{1}{\sqrt{5-4\cos\theta}}\right)\sin\theta=P(t)$$

其中，

$$\beta=\frac{2kL}{mg}\qquad P=\frac{M}{mgL}\qquad \tau=t\sqrt{\frac{g}{L}}\qquad \alpha=(c/m)\sqrt{L/g}$$

t 为时间。

如果 $M=0,\beta=10,\alpha=0.1,\theta(0)=\pi/4,d\theta(0)/d\tau=0$,画出转角 θ 的曲线(θ 为 τ 的函数,且 τ 在 $0\leqslant\tau\leqslant50$ 范围内 1000 等分)并在一单独图表中画出 $\theta(\tau)$ 相对于 $d\theta(\tau)/d\tau$ 的图形。

5.15 一根均匀不可伸展的电缆,长为 L_0,单位长度重为 w,悬挂在两个固定点 $x=0$ 和 $x=L$ 之间($L<L_0$)。如果绳子没有任何弯曲刚度,且仅能支持拉力 T,那么电缆的无量纲挠度 $z(\eta)$ 的控制方程为：[2]

$$\frac{d^2z}{d\eta^2}=\beta\sqrt{1+\left(\frac{dz}{d\eta}\right)^2}$$

其中 $\eta=x/L,\beta=wL/H,H$ 是 T 的水平分量,负 z 表明向下偏移。相应电缆的长度 L_o 等于：

$$L_o=L\int_0^1\sqrt{1+\left(\frac{dz}{d\eta}\right)^2}d\eta$$

当 w、L 和 L_o 给出时可确定 β 和 H 值。其边界条件为：

$$z(0)=0\qquad z(1)=0$$

$L_0/L=1.2$ 时求 β 值和斜率 $dz(0)/d\eta$。求解方法要求使用嵌套的迭代循环：最内部的循环用以确定满足 $z(1)=0$ 的斜率 $dz(0)/d\eta$ 的值,最外部的循环用以确定满足积分的 β 值。因为求取斜率 $dz(0)/d\eta$ 要用到 β 值,所以要求嵌套。用 trapz 进行积分。

[答案：$\beta=2.1284,dz(0)/d\eta=-1.2768$。]

5.16 由一长管连接的两个矩形等截面水库,高度差为 Z,它所产生的振动由下式确定：[3]

$$\frac{d^2Z}{dt^2}+\mathrm{signum}(dZ/dt)p\left(\frac{dZ}{dt}\right)^2+qZ=0$$

如果 $p=0.375\ \mathrm{m}^{-1},q=7.4\times10^{-4}\ \mathrm{s}^{-2}$,初始条件为 $Z(0)=Z_n\mathrm{m},dZ(0)/dt=0$ m/s,求当 $Z_1=10$ m,$Z_2=50$ m 时,第一次出现 $Z(t_n)=0$ 的 $t_n(n=1,2)$ 值。用 interp1 求 t_n,数值符号由 sign 确定。建议:绘制一个 Z_n 值的结果,然后根据曲线的特性,使用 min 和 find 的适当组合来选择小范围值的中值索引，interp1 利用它可实现插值。

---

① 参见 H.B.Wilson 和 L.H.Turcotte,出处同上,第 279 页。

② 参见 M.Irvine,出处同上,第 4 页。

③ 参见 D.N.Roy, *Applied Fluid Mechanics*, Ellis Horwood Limited, Chichester, England, 1988 年,第 290－293 页。

[答案：$t_1 = 114.269\,2$ s，$t_2 = 276.142\,8$ s。]

关于第 5.5.5 节内容的练习：

5.17 (a)用 fsolve 搜索以度数表示的 θ 值和 k 值并使其在 $a = 1, b = 3$ 时满足下列方程：

$$b = k(1 - \cos\theta)$$
$$a = k(\theta - \sin\theta)$$

(b) (a)中的两个方程可合并为下面的一个方程：

$$b(\theta - \sin\theta) - a(1 - \cos\theta) = 0$$

当 $a = 1, b = 3$ 时，用 fzero 求 θ 值，然后用(a)中的一个方程求取 k。

[答案：$k = 6.918\,9, \theta = 55.499\,9°$。]

5.18 (a) $\sigma = 5.667 \times 10^{-8}$，$T_1 = 373$ K，$T_2 = 293$ K 时，用 fsolve 从下述方程求取 Q, T_A, T_B 值。

$$T_1^4 - T_A^4 = Q/\sigma$$
$$T_A^4 - T_B^4 = Q/\sigma$$
$$T_B^4 - T_2^4 = Q/\sigma$$

(b) (a)中方程也可写为：

$$\begin{bmatrix} 1 & 0 & 1/\sigma \\ 1 & -1 & -1/\sigma \\ 0 & 1 & -1/\sigma \end{bmatrix} \begin{Bmatrix} x \\ y \\ Q \end{Bmatrix} = \begin{Bmatrix} T_1^4 \\ 0 \\ T_2^4 \end{Bmatrix}$$

其中 $x = T_A^4, y = T_B^4$。使用左除从方程组中求取 Q, T_A, T_B 值。

[答案：$T_A = 352.052, T_B = 326.511\,6, Q = 226.431\,2$。]

关于第 5.6.1 节内容的练习：

5.19 如图 5.15 所示，阶跃循环轴的应力集中系数可由下式近似估算：[①]

$$K_t = c\left(\frac{D - d}{2d}\right)^{-a}$$

其中 c 和 a 在表 5.3 中给出。通过下述两种方法求取 c 和 a 的两个表达式(D/d 的函数)：(1)用五阶多项式，(2)用 spline 函数。对于上述两种方法，比较由两组拟合值获得的 K_t 值和表 5.3 中给出的初始值。指出哪一个是这种情况下较好的一种方式。

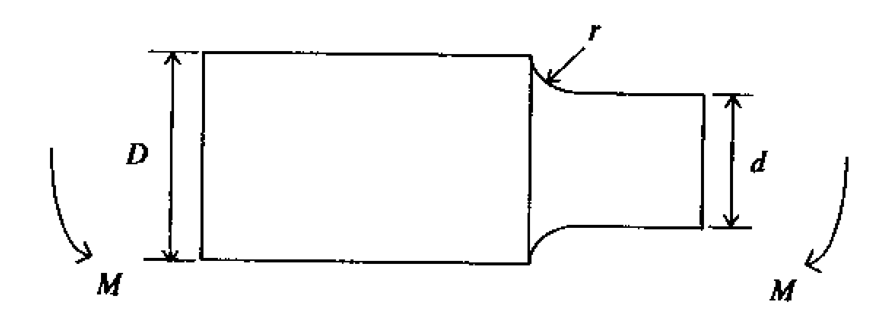


图 5.15 应力集中系数的几何图形和负荷

① 参见 R.L.Norton, *Machine Design, An Integrated Approach*, Prentice Hall, Upper Saddle River, NJ, 1996 年，第 1005ff 页。

表5.3 应力集中系数常数

D/d	c	a
6.00	0.88	0.33
3.00	0.89	0.31
2.00	0.91	0.29
1.50	0.94	0.26
1.20	0.97	0.22
1.10	0.95	0.24
1.07	0.98	0.21
1.05	0.98	0.20
1.03	0.98	0.18
1.01	0.92	0.17

关于5.6.4节内容的练习：

5.20 考虑如下信号：

$$f(t)=\sum_{n=1}^{4}H_n e^{-\xi_n\omega_n t}\sin\left(\sqrt{1-\xi_n^2}\omega_n t\right) \qquad 0\leqslant t\leqslant T$$

其中常数值由表5.4给出。对于 $N=2^{10}$和 $\Delta t=2\pi/(4\omega_{:4})$：

(a)分别对带有和不带有汉明权函数的信号画出幅度频谱。结果如图5.16所示。

(b)确定峰值出现时的频率。[提示:用 find 和 diff 的几个应用。]

[答案:不带有汉明权函数:[4.84375,9.14063,20.0781]Hz,带有汉明权函数:[4.92188,9.0625,9.45313,20.0781]Hz。]

表5.4 定义练习5.20中信号的常数

n	$\omega_n/2\pi$	ξ_n	H_n
1	5	0.1	1
2	9	0.04	1.3
3	9.4	0.04	1.3
4	20	0.03	1.8

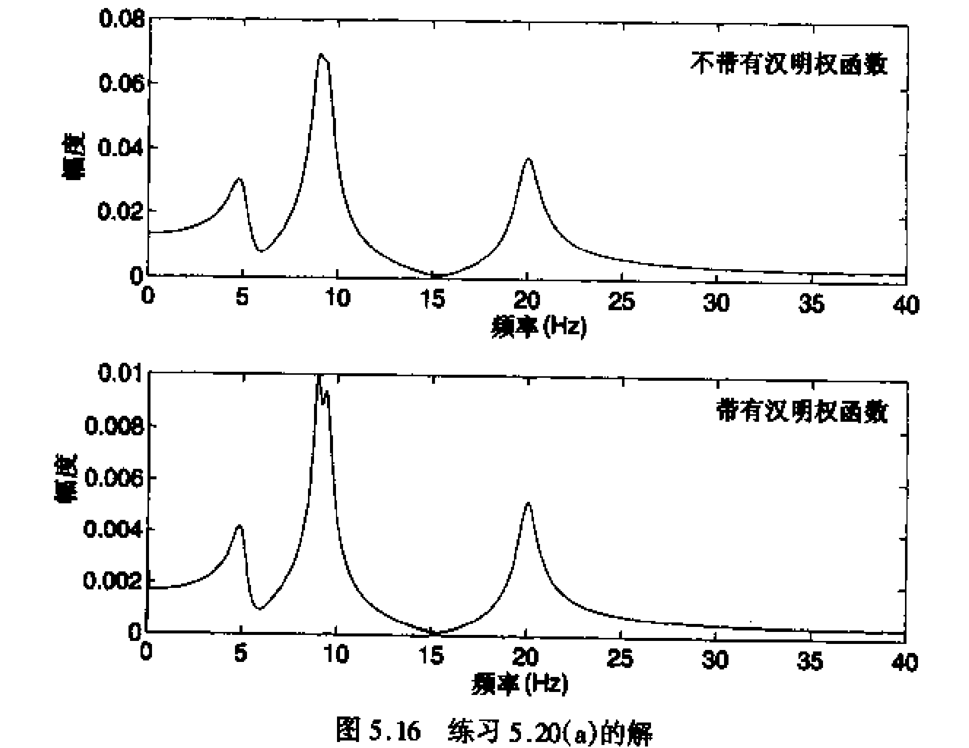


图5.16 练习5.20(a)的解

第6章 二维图形

本章介绍了许多二维绘图功能的实现方法。

6.1 概述

MATLAB 提供了很多灵活易用的二维和三维绘图功能函数。这些绘图函数分为三类:图形处理、曲线和曲面图的创建、注释和图形特性。绘图函数虽多,但语法大致相同,可归为同类函数。本章和下章用到的函数有:

管理类	生成类	注释和特性类
figure	2-D	xlabel
subplot	plot	ylabel
zoom	polar	zlabel(仅适用于3D)
hold	fill	text
view(仅适用于3D)	plotyy	text3(仅适用于3D)
rotate3d(仅适用于3D)	3-D	title
	plot3	legend(仅适用于2D)
	surf,surfc	box
	mesh,meshz	set
	contour,contour3	grid
	contourf	axis,axis equal,
	waterfall	axis off
	cylinder	colorbar (仅适用于3D)
		clabel
		colormap (仅适用于3D)

其他特殊绘图函数,如 bar 和 hist 函数,将在以后章节中介绍。

生成图形实体时,应确保做到以下两点:(1)突出重点以满足解的客观性;(2)通过使用坐标轴标注、图形标题、曲线标注(如有多条曲线)及重要数值标注方法使其清晰易懂、特点明显。而且,对强调功能的颜色、线型、符号和文本在不影响效果的前提下也应加以使用。

一组典型的生成图形表达式包括处理函数,后面是一个或多个图形生成函数,然后是注释函数,可能其后还有附加的管理函数。除了管理函数,其余函数可以采用任意的顺序。而且,注释函数和图形属性函数是可选的。MATLAB 在坐标轴上标出刻度及刻度值,即使多值输出也如此。只要能得到部分标注的图形,就证明函数语法的使用是正确的。

当图形的管理、生成、注释和属性函数被调用时,图形将显示在图形窗口中,该窗口由 MATLAB 在运行时生成。当一段程序(函数)用到几个图形函数时,MATLAB 将生成一个新的图形窗口。然而,在生成新的图形窗口之前,任何先前生成的图形窗口将被移走。为在其自己的图形窗口显示每一幅新图形,必须使用函数:

```
figure(n)
```

其中 n 是整数。如果省略 window 参数，MATLAB 给出下一个整数值。

也可以用：

```
subplot(i,j,k)
```

把几个独立生成的图形放在同一图形窗口中。前两个参数把窗口分为不同的块(行和列)，第三个参数指出图形放在第几个块中，值取 1 表示左上角的块，值等于行数和列数之和时表示应放于右下角的块。随着数字的增大，显示的部分按从左到右，从上到下的顺序定位。在程序中任何一个出现在 figure/subplot 之后的注释函数和管理函数仅适用于 subplot 函数第 3 个参数指定的块中。每一块中，均可使用二维和三维图形生成函数。参考图 6.1 中使用 figure 和 subplot 的几个例子。由图中可看出如果仅需要一个图形窗口，则即使用到了 subplot 函数，figure 也可以省略。

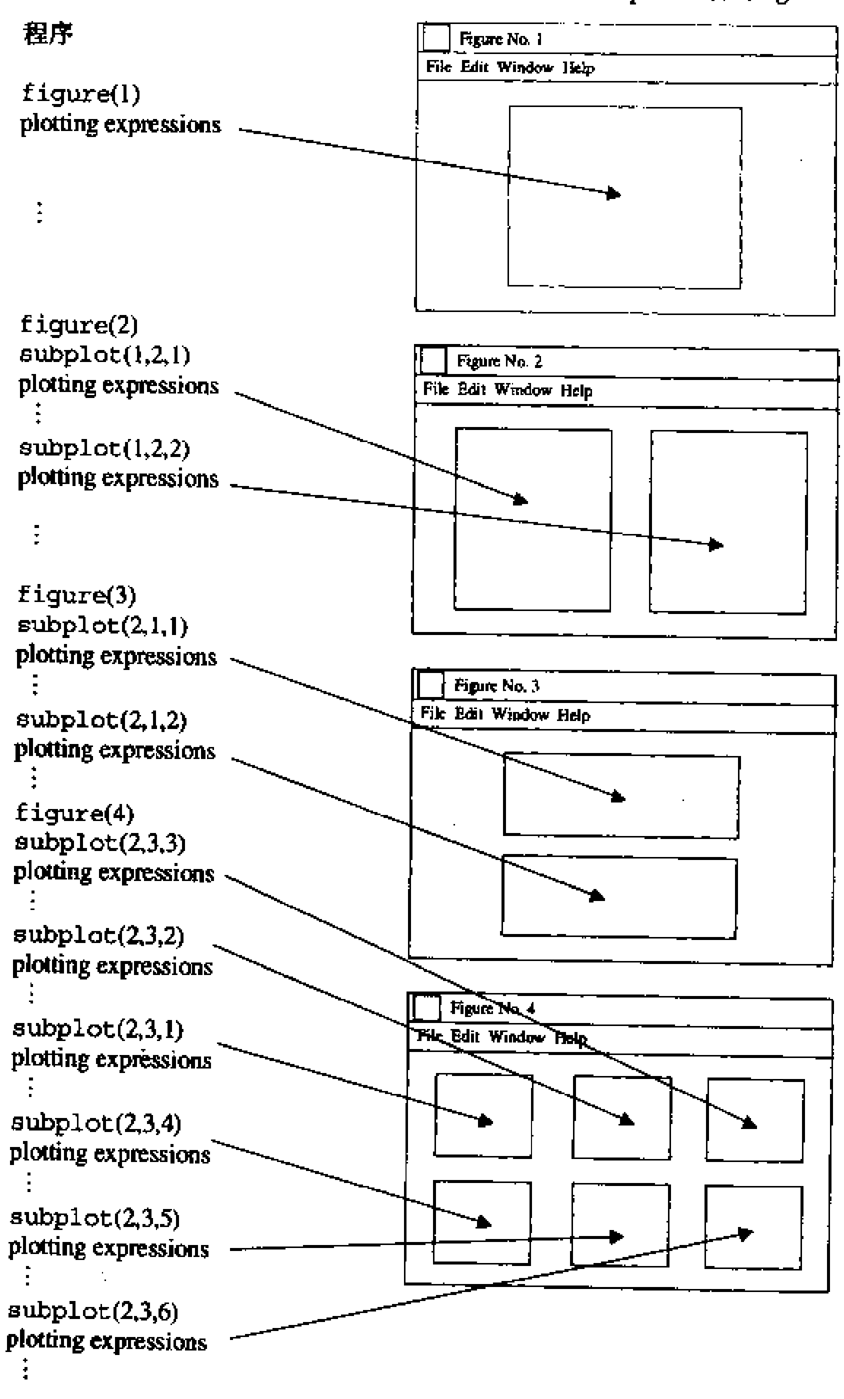


图 6.1　figure 和 subplot 不同组合应用示例

由于每一个图形生成函数创建一个新的图形窗口[①]，为了在一个图形中画出多条曲线、多个曲面图或线(或它们的组合)，必须使用：

```
hold on
```

该语句保持当前窗口(或 subplot 子块)为激活状态，但必须使用一组兼容的图形创建函数，如 surf 与 plot3 或 plot 与 fill。在每个图形窗口中，所有图形均可通过 Edit 下拉菜单 Copy Figure 命令复制到 Windows 剪切板。图形可被传输(粘贴)到字处理程序的页中并保存为 Windows 图元文件格式。

MATLAB 提供了将图形转换为与多种普通打印设备相兼容的格式的方法。例如，如果把当前活动图形窗口的图形存为二层压缩文件，名为 FileName，然后在黑白打印机上输出，可使用下述语句：

```
print -deps2 'c:\path\FileName.eps'
```

其中 *path* 指明文件所在的目录和子目录。其他选项参见 print 的帮助文件。另一方面，如果想将二层压缩文件插入到 MS Word 文档，以显示图形的"tiff"预览图像，可以用下述语句：[②]

```
print -deps2 -tiff 'c:\path\FileName.eps'
```

6.2 基本二维绘图命令

基本二维绘图命令为：

```
plot(u1,v1,u2,v2,c2...)
```

其中 u_j 和 v_j 分别是某点或一系列点的 x 和 y 坐标。它们或者是成对的数字、相同长度的向量、同阶次矩阵，或是计算时生成上述三者之一的表达式。c_j 是字符串：一个字符确定画线/点的颜色，另一个字符确定画点的类型，两个字符用于定义线的特征。要画一系列点时，c_j 可以是's'(用方框画点)或是'*'(用星号画点)。不论是否显示，都应用(直)线将这些点连接起来，字符 c_j 可以是'-'，代表实线；也可以是'--'，代表虚线。当以相同的颜色绘制线和点时，c_j 包含两种描述符。例如，要用蓝色虚线连接蓝色菱形点时，c_j 为'*b -- d*'，单引号内三个字符的顺序不重要。当点和线一起画，而且定义线的点与要画的点数目不同时，c_1 定义线型符号，c_2 定义画点符号，反之亦然。颜色与线型及点型的符号定义参考 plot 的帮助文件。如果省略 c_j，则使用系统默认值。如所画曲线多于一条，则曲线颜色按默认顺序变化。

下面给出的方法可以画出点、线、圆、表达式、曲线族和多个函数描述的曲线。

6.2.1 画点

在(2,4)坐标处加一红色星号，其语句为：

```
plot(2,4,'r*')
```

---

① MATLAB 窗口的外观、管理和文件管理的描述依赖于 Windows 环境。其他操作系统使用过程与之类似。

② 为在 MS Word 中使用该文件，必须要安装相应的压缩过滤器。过滤器作为 MS Word 的一部分，但却不是 MS Word 的默认安装。本例中，可运行 MS Word 的安装程序并安装相应的过滤器。

6.2.2 画线

用默认线型(实线)及默认颜色(蓝色)画一条从(0,0)到(1,2)的直线,语句为:

```
plot([0 1],[0 2])
```

第 1 个二元向量[0 1]为 x 坐标值,第 2 个二元向量[0 2]为 y 坐标值。因此,每一向量的第 1 个元素定义了线段起点的(x,y)坐标,第 2 个元素定义了终点的坐标。

设想画 n 条互不相连的线段,每条线段的两个端点坐标分别是(x_{1n},y_{1n})和(x_{2n},y_{2n})。为此创建 4 个向量:

$$\boldsymbol{x}_j=[x_{j1},x_{j2}\ldots x_{jn}]$$
$$\boldsymbol{y}_j=[y_{j1},y_{j2}\ldots y_{jn}] \qquad j=1,2$$

则 plot 指令为:①

```
plot([x1;x2],[y1;y2])
```

其中[$x1;x2$]和[$y1;y2$]都是($2\times n$)阶矩阵。

为说明这个表达式,在 $x=2,4,6$ 和 8 处画 4 条从 $y=0$ 到 $y=\cos(\pi x/20)$ 的垂直线。程序如下:

```
x=2:2:8
plot([x;x],[zeros(1,length(x));cos(pi*x/20)],'k')
```

由于 $x_1=x_2$,因此所有线段都具有相同的颜色(本例中为黑色)。函数 zeros 用于创建与 x 等长的 0 向量。结果如图 6.2(a)所示。但是,由于 MATLAB 会自动标出轴线刻度,使得第一条和最后一条直线与图形的边轴重合,而变得不可见。为此,可使用如下语句对轴进行调整以使直线可见。

```
axis([xmin xmax ymin ymax])
```

其中 $x_{min},x_{max},y_{min},y_{max}$ 分别是 x 轴和 y 轴的最小和最大值。修改后的程序为:

```
x=2:2:8;
plot([x;x],[zeros(1,length(x));cos(pi*x/20)],'k')
axis([1 9 0 1])
```

修正图形如图 6.2(b)所示。

得到轴的边界值,并根据情况对其进行重新定义可以提供更大的灵活性。边界值可由下述语句获得:

```
v=axis;
```

其中 $\boldsymbol{v}$ 是一个四元向量:

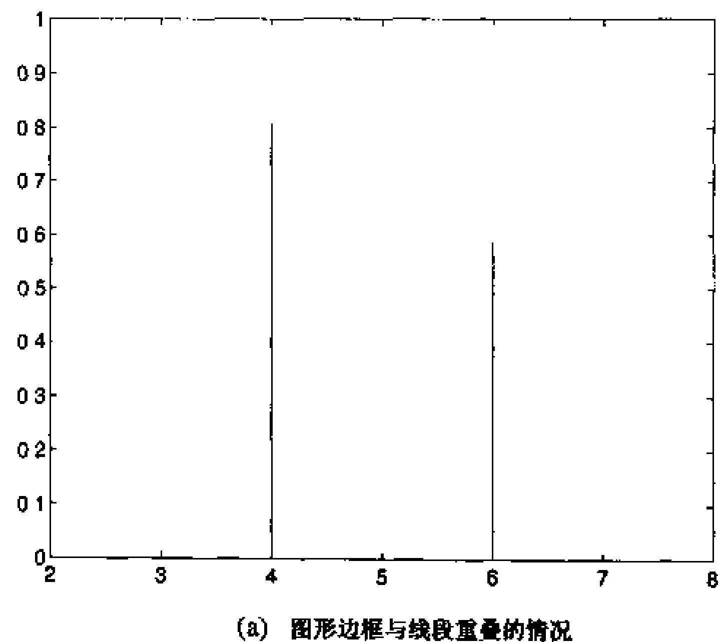
$$\boldsymbol{v}(1)=x_{\min} \qquad \boldsymbol{v}(3)=y_{\min}$$
$$\boldsymbol{v}(2)=x_{\max} \qquad \boldsymbol{v}(4)=y_{\max}$$

因此,为获得如图 6.2(b)所示的修正图形,程序可进一步修改如下:

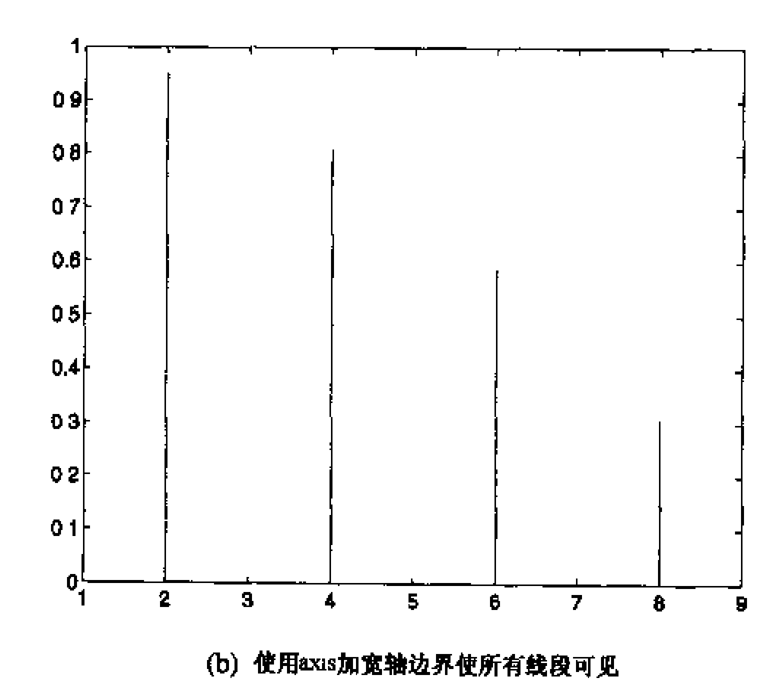
```
x=2:2:8
```

① 当 $x_1=x_2,y_1=0$ 时,plot 表达式是 stem 函数的概括。

```
plot([x;x],[zeros(1,length(x));cos(pi * x/20)],'k')
v = axis;
v(1) = 1;
v(2) = 9;
axis(v)
```




(a) 图形边框与线段重叠的情况



(b) 使用axis加宽轴边界使所有线段可见

图 6.2 图形与边界

6.2.3 画圆

在笛卡儿坐标系中画一个半径为 r,圆心坐标为(a,b)的圆,首先应进行如下坐标变换(见图 2.2)。

$$x = a + r\cos(\theta)$$

$$y = b + r\sin(\theta)$$

其中 $0 \leqslant \theta \leqslant \theta_l \leqslant 2\pi$。当 $\theta_l < 2\pi$ 时,图形为一段圆弧。设 $\theta_l = 2\pi$, $a = 1$, $b = 2$, $r = 0.5$,则画圆的程序如下:

```
theta = linspace(0,2 * pi);
plot(1 + 0.5 * cos(theta),2 + 0.5 * sin(theta))
axis equal
```

函数 axis equal 使图形对称,因而图形是圆形,而不是椭圆。

画出6个同心圆,其初始半径为0.5,半径增量为0.25,圆心处标上加号,程序如下:

```
theta = linspace(0,2 * pi,50);          %(1×50)
rad = 0.5:0.25:1.75;                    %(1×6)
x = 1 + cos(theta)' * rad;              %(50×6)
y = 2 + sin(theta)' * rad;              %(50×6)
plot(x,y,'k',1,2,'k + ')
axis equal
```

按列画出矩阵中的值。因为对应每个 *rad*(弧度)值要画出50个 θ 值,所以定义矩阵阶次为(50×6)。如果省略字符串'k',则每个圆的颜色不同。程序执行后生成图6.3。

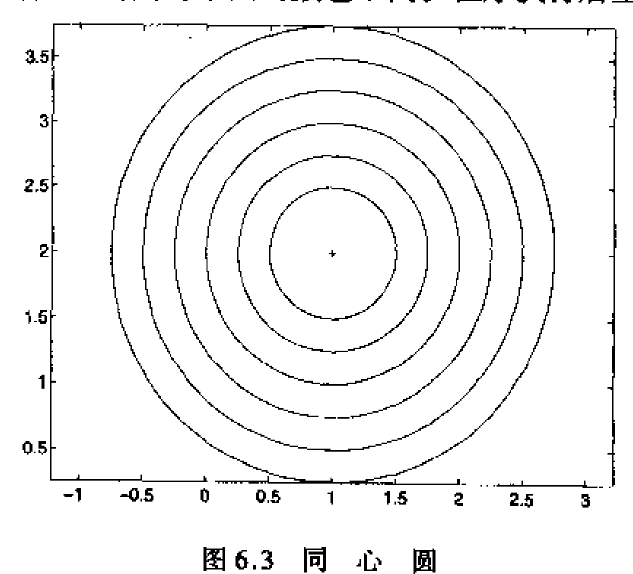


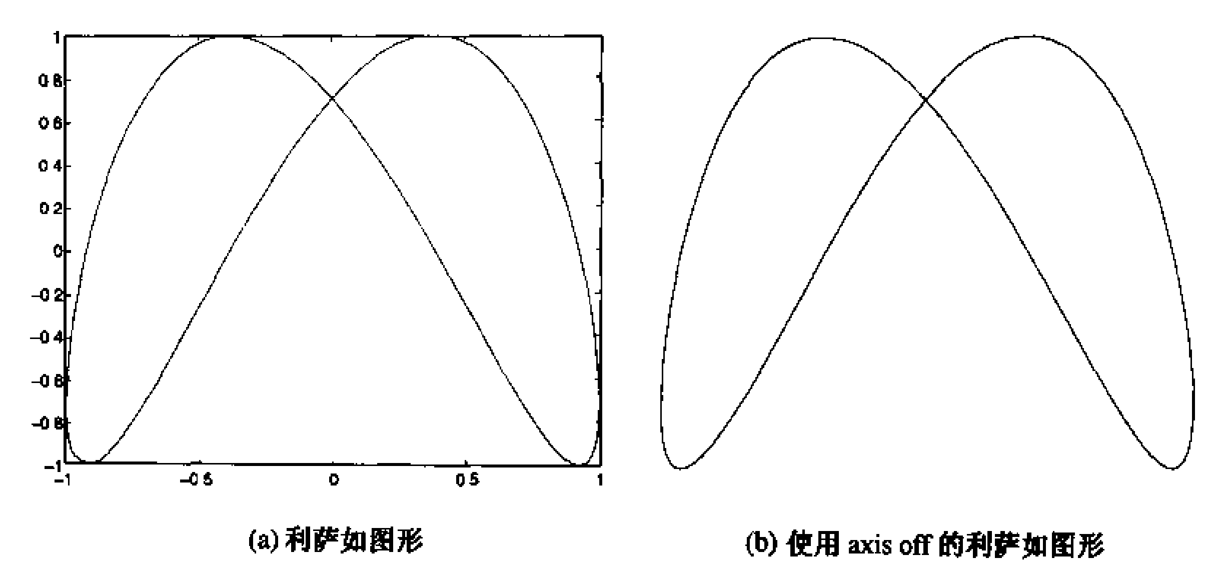
图6.3 同心圆

6.2.4 函数相对另一函数的图形绘制

在坐标图上画出 $\sin(n\theta)$ 对 $\sin(m\theta + \theta_1)$ 的关系图就得到利萨如(Lissajous)图形,其中 m, n 是正数,$0 \leqslant \theta \leqslant 2\pi$,$0 \leqslant \theta_1 < 2\pi$。设 $n = 1$, $m = 2$, $\theta_1 = \pi/4(45°)$。如将 θ 进行101等分,则程序如下:

```
theta = linspace(0,2 * pi,101);
plot(sin(theta),sin(2 * theta + pi/4))
```

程序执行后生成图6.4(a)。也可在程序中加入 axis off 语句得到不带参考轴的利萨如图形,结果如图6.4(b)所示。



(a) 利萨如图形　　(b) 使用 axis off 的利萨如图形

图 6.4　利萨如图形示例

6.2.5　曲线族的绘制

前面绘制 6 个同心圆的例子中给出了一种绘制曲线族的方法。一般来说，MATLAB 允许用向量表示一个轴，用矩阵表示另一个轴，并根据矩阵的行和列与向量长度的匹配情况画出向量对匹配的行或列的曲线图。

画抛物线族的情况如下：

$$y = a^2 - x^2$$

其中 $-5 \leqslant x \leqslant 5, a = 1,2,\cdots, 5$，程序如下：

```
x = -5:0.2:5;
a = 1:5;
[xx,aa] = meshgrid(x.^2,a.^2);
plot(x,aa - xx,'k')
```

结果如图 6.5 所示。

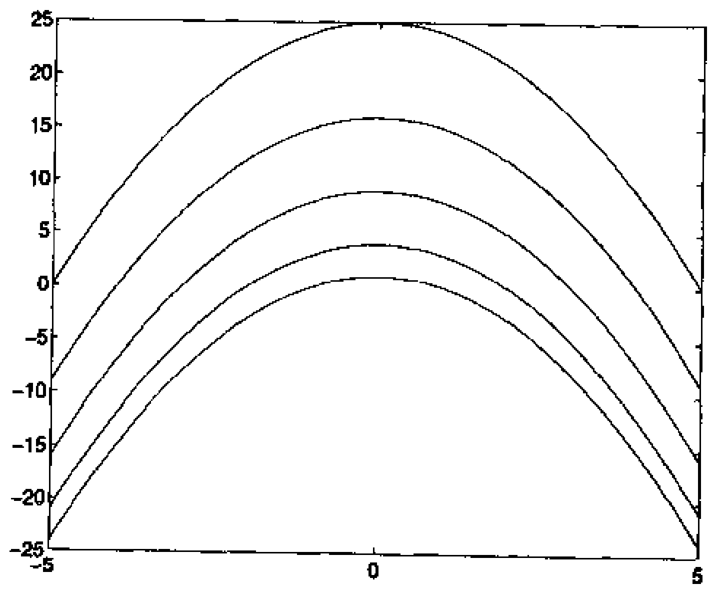


图 6.5　抛 物 线 族

考虑级数收敛的图形表示：

$$S_N = \sum_{j=1}^{N} \frac{1}{(a+j)^2}$$

其中 $N=1,2,\cdots,10$ 并且 $a=1,2,3$。该例中，应用 cumsum 函数（见第 2.5 节），程序如下：

```
aa = 1:3;                       %(1×3)
N = 1:10;                       %(1×10)
[a,k] = meshgrid(aa,N);         %(10×3)
S = cumsum(1./(a+k).^2);        %(10×3)
plot(N,S,'K')
```

结果如图 6.6 所示。

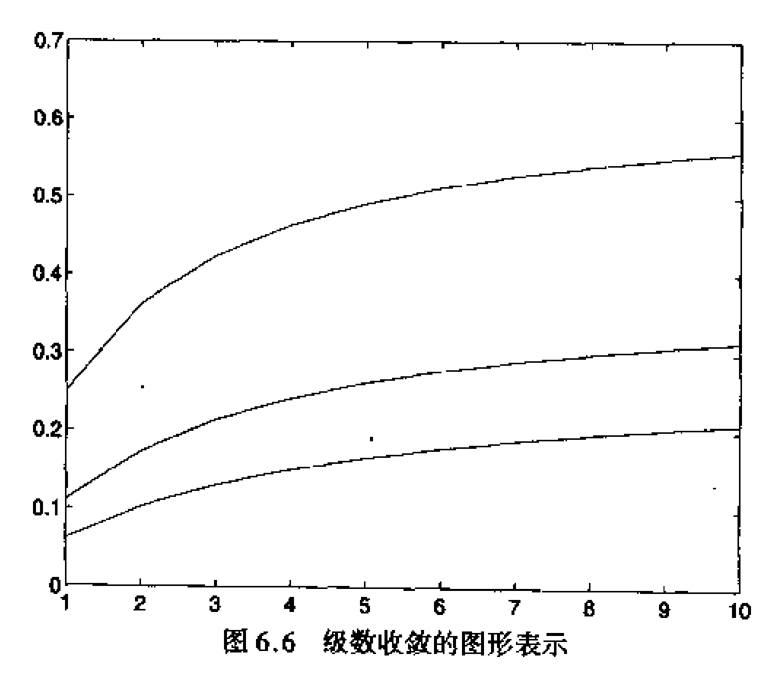


图 6.6 级数收敛的图形表示

6.2.6 在一个图形界面中绘制多个函数图形[①]

考虑如下三个函数：

$$g_1(x) = 0.1x^2$$

$$g_2(y) = \cos^2 y$$

$$g_3(z) = e^{-0.3z}$$

其中 $0 \leqslant x = y = z \leqslant 3.5$。可用下述三种方法中的任何一种画出这三条函数曲线：

```
x = linspace(0,3.5);
plot(x,[0.1*x.^2;cos(x).^2;exp(-0.3*x)],'k')
```

或

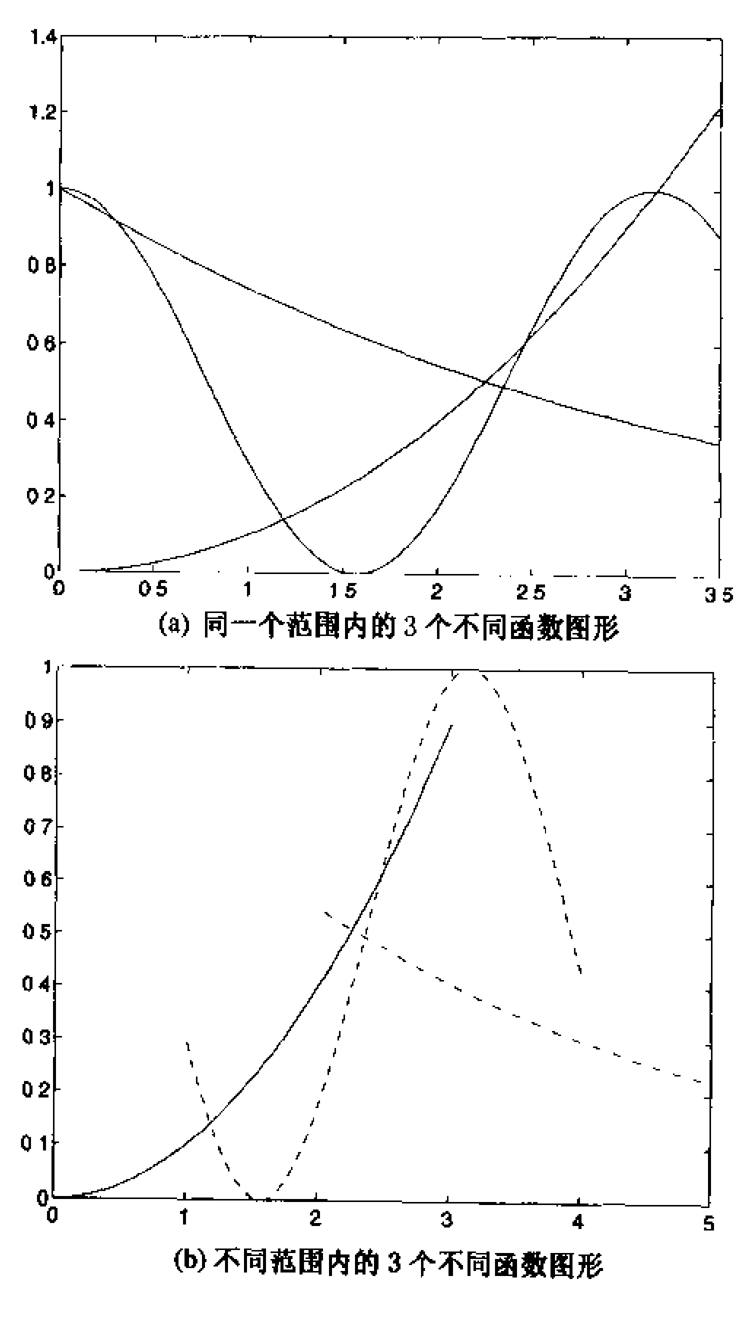
```
x = linspace(0,3.5);
plot(x,0.1*x.^2,'k',x,cos(x).^2,'k',x,exp(-0.3*x),'k')
```

或

① 用 plotyy 在两种不同纵坐标内绘制两种不同类型图形。参见第 6.3.6 节所述及生成图 14.1 的程序。

```
x = linspace(0,3.5);
plot(x,0.1 * x.^2,'k')
hold on
plot(x,cos(x).^2,'k')
plot(x,exp(-0.3 * x),'k')
```

执行任一程序均可生成图 6.7(a),其中所有曲线均为黑色。



(a) 同一个范围内的 3 个不同函数图形

(b) 不同范围内的 3 个不同函数图形

图 6.7 函数图形

另一方面,如果这些函数中自变量的范围不同,则只有第 2 段和第 3 段程序可用。例如,如果 $0\leqslant x\leqslant 3, 1\leqslant y\leqslant 4, 2\leqslant z\leqslant 5$,则第 2 段程序使用如下:

```
x = linspace(0,3,45);
y = linspace(1,4,55);
z = linspace(2,5,65);
plot(x,0.1 * x.^2,'k-',y,cos(y).^2,'k--',z,exp(-0.3 * z),'k-.')
```

结果如图6.7(b)所示。注意绘制每个函数图形时使用了不同线型，且绘制每条曲线所使用的点数不同。

6.3 图形注释和可视化提高

6.3.1 坐标轴和曲线的标注、图形标题、图注、文本和其他属性

举例说明如何提高图形的可视化性能：

- 使用轴标注、图形标题、曲线标注、图注、填充区域和添加文本
- 改变轴、曲线和文本属性
- 使用希腊字母、数学符号和上下标

画出标注、标题并说明两条相交曲线 cos(x)和 1/cosh(x)的关系，其中$0 \leq x \leq 6$。在该范围内，两条曲线在$x=4.73$处相交。参考第5.5.1节中的例子。再画一条过交点的垂直线，并在相交点附近标出x值。程序如下：

```
x=0:.05:6;
plot(x,cos(x),'k',x,1./cosh(x),'k',[4.73 4.73],[-1 1],'k')
xlabel ('x')
ylabel('Value of functions')
title('Visualization of two intersecting curves')
text(4.8,-.1,'x=4.73')
text(2.1,.3,'1/cosh(x)')
text(1.2,-.4,'cos(x)')
```

结果如图6.8所示。plot 函数执行后即已确定出标注文本所在的坐标值——也就是说，仅在写出程序前两句并给出结果图形之后方可确定标注文本的坐标值。然后是 text 函数语句。

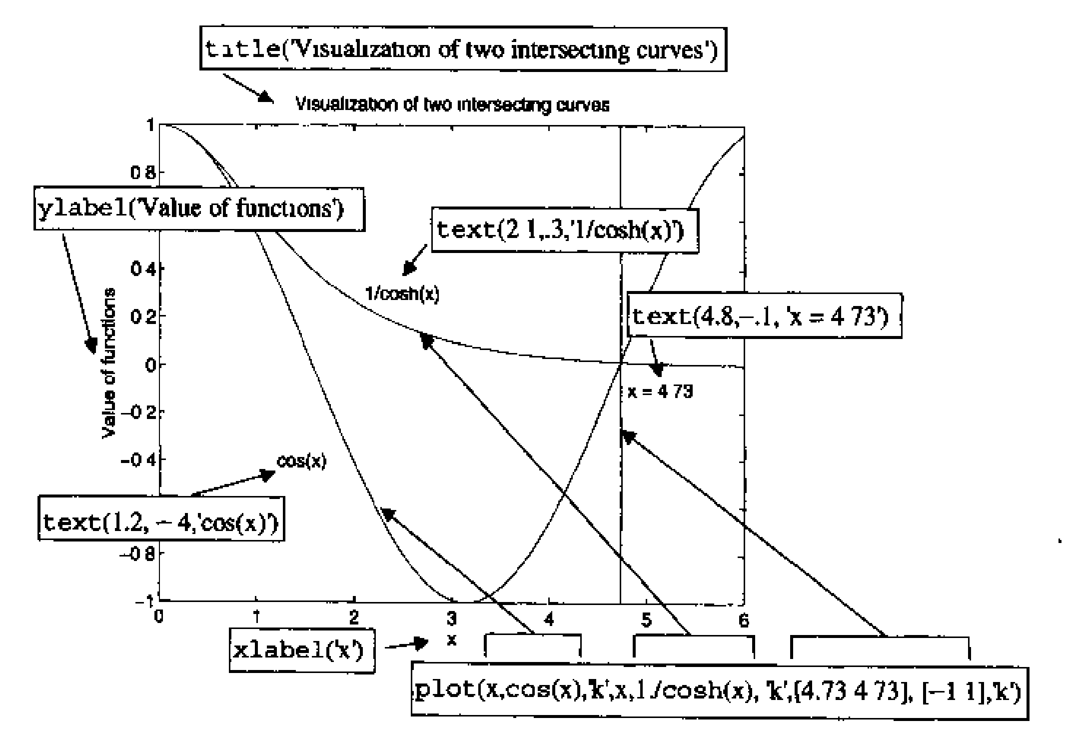


图6.8 创建及标注所示图形的表达式

可以修改上述结果，使两条曲线在 $0\leqslant x\leqslant 4.73$ 范围内的区域填充为蓝绿色。填充两条曲线之间的区域采用如下语句：

```
fill
```

它要求这两条曲线连成一个多边形，首先创建一个新的范围 $0\leqslant x\leqslant 4.73$。为此，在上述程序中添加如下语句：

```
xn = linspace(0,4.73,50);
hold on
fill([xn fliplr(xn)],[1./cosh(xn) fliplr(cos(xn))],'c');
```

修改后程序执行结果如图 6.9 所示。创建向量[1./cosh(xn) fliplr(cos(xn))]，则可生成相连的多边形，它是由顶部曲线 1/cosh(x)和底部曲线(cos(x)的逆向量)相互连接而成。新的 x 坐标向量 [xn fliplr(xn)]对应新向量，它由 x 的新值和它的逆组合而成。

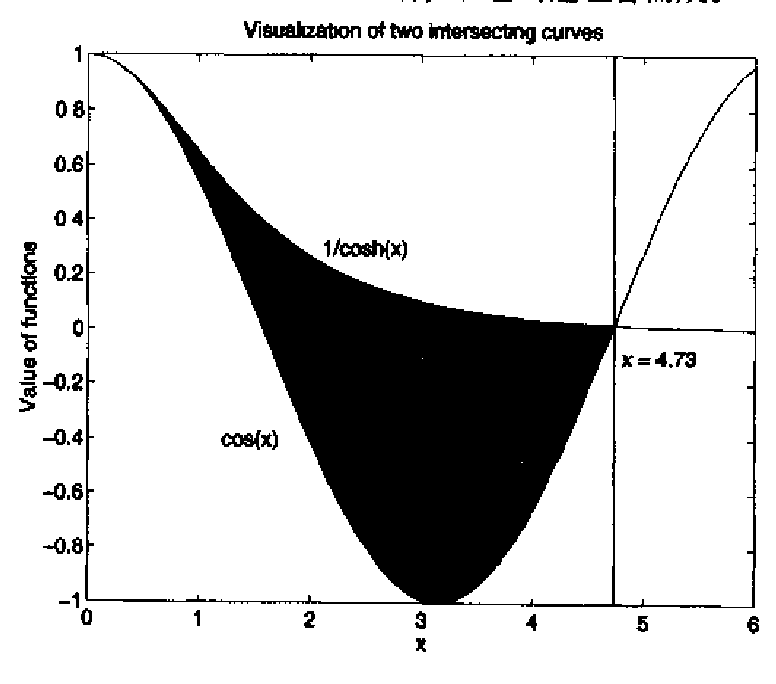


图 6.9 填充图 6.8 中曲线间区域

如果不使用填充命令，而用 20 条等间隔的垂直线填充，程序如下：

```
x = 0:.05:6;
plot(x,cos(x),'k',x,1./cosh(x),'k',[4.73 4.73],[-1 1],'k')
hold on
xx = linspace(0,4.73,20);
plot([xx;xx],[cos(xx);1./cosh(xx)],'k-')
```

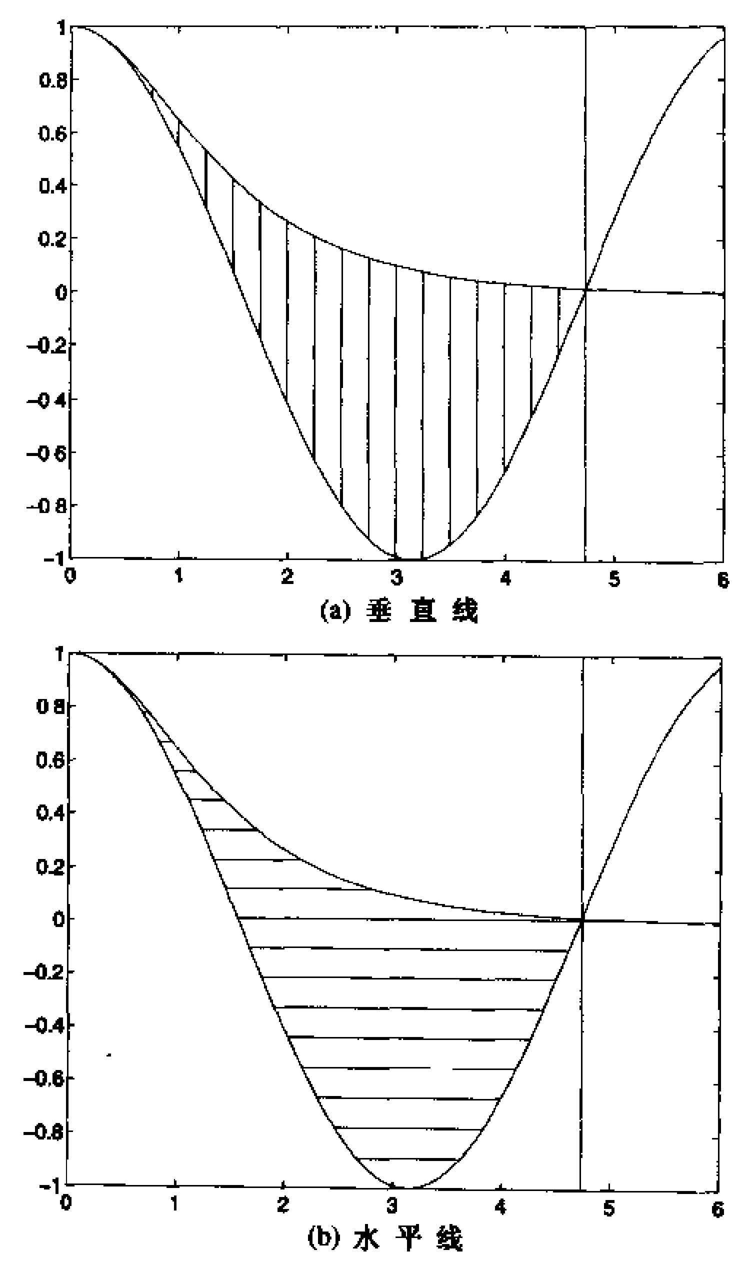
结果如图 6.10(a)所示。

如果想画出 20 条等间隔的水平线，必须用反函数 arccos(x)和 arcosh(x)。程序如下：

```
x = 0:.05:6;
plot(x,cos(x),'k',x,1./cosh(x),'k',[4.73 4.73],[-1 1],'k')
hold on
y1 = linspace(1,0.01,10);
plot([acos(y1);acosh(1./y1)],[y1;y1],'k-')
y2 = linspace(0.01,-1,10);
```

```
plot([acos(y2);pi+fliplr(acos(y1))],[y2;y2],'k-')
```

结果如图6.10(b)所示。这两个程序结合起来可生成如图6.11所示的阴影线效果。



(a) 垂 直 线

(b) 水 平 线

图6.10 两条曲线相交区域用等间距线填充

还有另一种修改图6.9的方法,即使用:

```
legend
```

legend函数不同于text函数,因为text函数可多次使用,而legend函数却只可用一次。legend函数的参数个数与一个或多个plot函数绘制的不同线段个数相等。除此之外,legend函数还有一个可选(最后一个)参数。除了最后一个参数外,每一个参数都是一个字符串,代表每条线的字母数字标识符。可选参数的值(1,2,3或4),将图注放在图形四个角中的一处,放在图形右边取值(-1),最好的位置是值(0)——即对数据影响最小的位置。当可选项省略时,图注放在默认位置,即右上角。但也可以省略该参数而用鼠标来放置图注。其方法是用鼠标点中图注,按下鼠标键将其拖至目标处。但这种放置仅是临时的,当图形关闭后下次重建时,图注仍显示


在默认位置，则必须照此方式再重新定位。

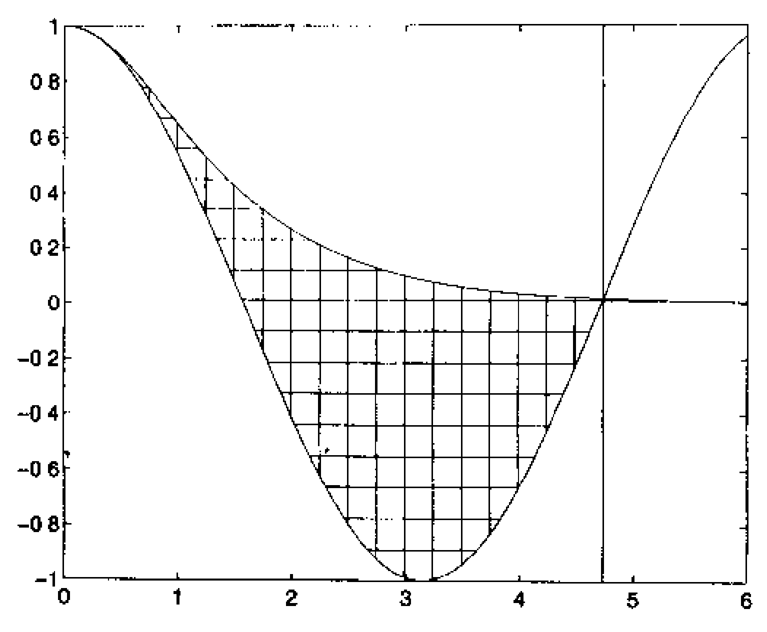


图 6.11 在两条曲线相交区域填充后阴影效果的图形

下面以生成图 6.8 的程序为例，通过观察其中各量位置的变化来说明 legend 函数的用法。参考下述语句：

```
x=0:.05:6;
plot(x,cos(x),'k-',x,1./cosh(x),'k--')
legend('cos(x) ','1/cosh(x)',3)
```

上述程序生成图 6.12。plot 函数中的两个三元组参数中的第三个参数确定了曲线为黑色，且 cos(x)曲线为实线，而 1/cosh(x)曲线为虚线。legend 参数与顺序相关，第一个参数与所画的第一条曲线相关，第二个参数与所画的第二条曲线相关，依此类推。如果有几个 plot 函数，则第二个 plot 函数的第一个参数在上一个 plot 函数的最后一个字符串之后，该字符串标识该函数中的最后一条曲线。在 figure 或 subplot 之后只能使用一次 legend 函数。legend 函数中的数字 3 将图注置于左下角。

图 6.12 也可由如下程序获得：

```
x=0:.05:6;
plot(x,cos(x),'k-')
hold on
plot(x,1./cosh(x),'k--')
legend('cos(x)','1/cosh(x)',3)
```

当未在 plot 函数中指定线型和颜色时，legend 函数使用默认线型(实线)和默认颜色序列，即图注中的线段为不同颜色的实线。

MATLAB 能够改变图形元素的所有属性。这里对文本属性的字体和大小、轴数、轴和曲线的线宽进行说明，所有属性的详细解释见 HTML 版本的帮助文件。默认线宽为 0.5，轴标注和文本的默认字体大小为 10，默认字体名为 Helvetica。

再参见生成图 6.8 的程序，其部分语句如下：

```
x=0: 05:6;
plot(x,cos(x),'k',x,1 /cosh(x),'k',[4 73 4.73],[-1 1],'k')
text(4.8,-.1,'x=4.73')
text(2 1,.3,'1/cosh(x)')
text(1 2,-.4,'cos(x)')
```

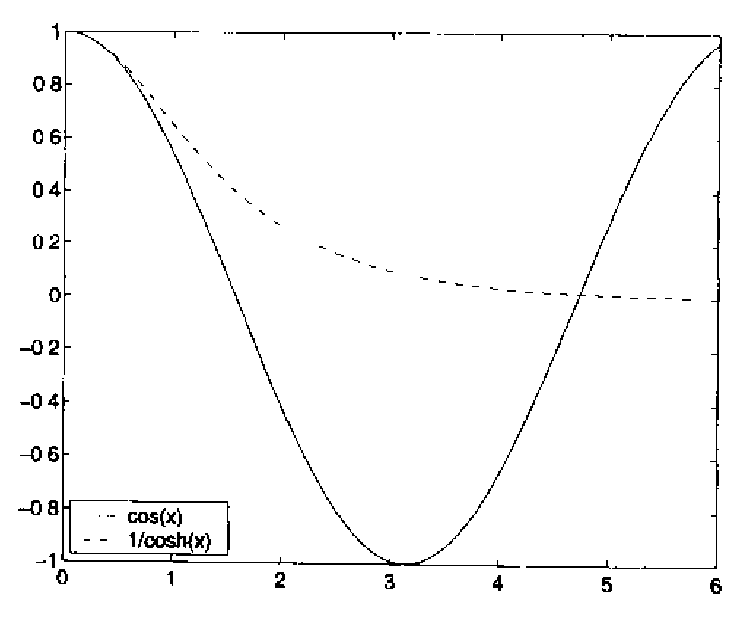


图 6.12 legend 函数的使用

现修改程序,以实现使轴线变粗,以不同宽度画出曲线,在左下角添加图注,用3种不同字号以及2种字型显示文本。修改后的程序为:

```
x=0:.05:6;
h=plot(x,cos(x),'k',x,1./cosh(x),'k',[4.73 4.73],[-1 1],'k');
text(4.8,-.1'x=4.73','fontname','times','fontsize',14)
text(2.1,.3,'1/cosh(x)','fontsize',16)
text(1.2,-.4,'cos(x)','fontsize',16,'fontname','times')
set(gca,'fontsize',14,'LineWidth',2)
PropertyName={'LineWidth','LineWidth','LineWidth'};
PropertyValue={2.5,2.5,2.5;7,7,7;1,1,1};
set(h,PropertyName,PropertyValue)
[legendhandle objecthandle]=legend('cos(x) ','1/cosh(x) ',3);
set(objecthandle(1),'fontsize','14,'color','r')
```

结果如图6.13所示。h = plot(…)在列向量 h 中放置3个值,h 叫做句柄,句柄标识了MATLAB要处理的图形对象(曲线)。本例中 h 有3个值,因为在该plot函数中要画三条不同的曲线: cos(x)、1/cosh(x)和 x = 4.73 处的一条垂直线。每条曲线按其所显示的顺序进行标号。用最后一个set函数设定每条线的线宽。第一个set函数使用MATLAB函数gca(= "get handle to current axis"的英文缩写),它返回轴的句柄。本例中,改变了轴的两个属性:线宽和字体大小。legend语句返回线属性和文本属性的句柄(*legendhandle* 和 *objecthandle*)。图注中文本属性的句柄由 *objecthandle*(1)访问,本例中用它来改变字体大小并使其颜色变为红色。

对文本、曲线和轴属性的修改也可以通过直接在图形窗口顶部选择适当的图标,或在Tools菜单中选择相应操作项来完成。图形一旦改变,可以保存。然而当生成该图形的程序或函数重新运行时,上述修改将不复存在而需重新设定。

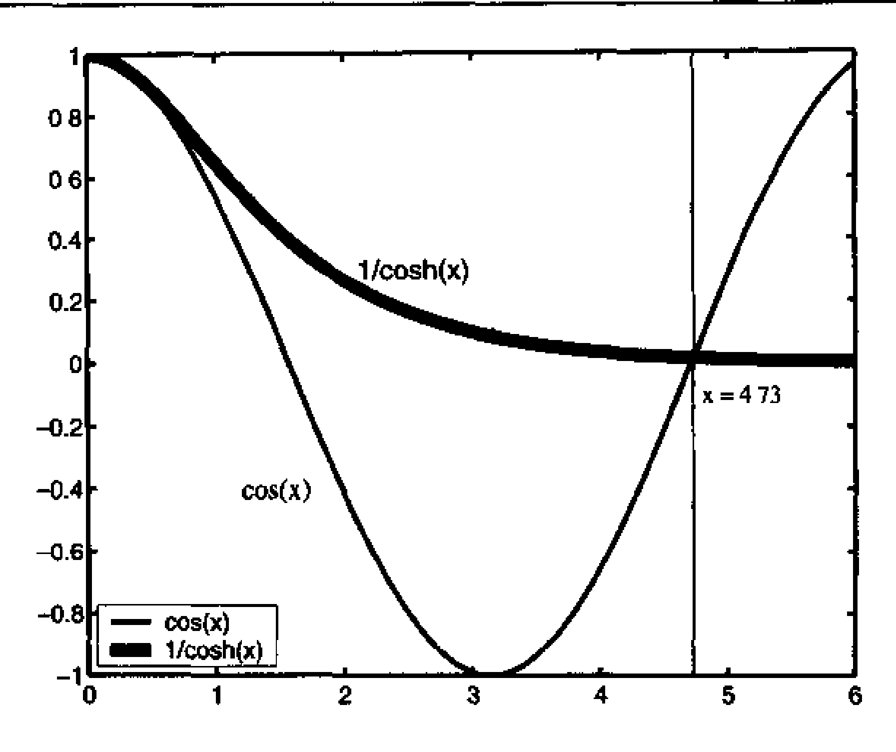


图 6.13 字体大小、字型和线宽的更改

另一个可由 text 使用的属性是 rotation，它可使文本相对于水平轴旋转 θ 度。第 6.3.6 节的图 6.20 将对该属性进行说明。

另外，还可以用大小写希腊字母、上标下标和数学符号对图形进行标注。标注可通过 xlabel、ylabel、zlabel、text、legend 和 title 完成。这些指令格式遵从 LaTeX 语言①，并包括前述文本修改命令。

下标由下划线(_)生成；上标由指数运算符(^)生成；希腊字母的生成如表 6.1 所示：先输入反斜杠(\ )，再输入字母的拼音。然而，因为许多大写的希腊字母与大写的英文字母相同，所以表中仅列出二者不同的部分，其他大写希腊字母可简单输入大写英文字母代替。

表 6.1 大小写希腊字母和部分数学符号

小写				大写		数学符号			
符号	句法	符号	句法	符号	句法	符号	句法	符号	句法
α	\ alpha	ν	\ nu	Γ	\ Gamma	$\leq$	\ leq	$\circ$	\ circ
β	\ beta	ξ	\ xi	Δ	\ Delta	$\geq$	\ geq	$\ll$	\ ll
γ	\ gamma	o	o	Θ	\ Theta	$\neq$	\ neq	$\gg$	\ gg
δ	\ delta	π	\ pi	Λ	\ Lambda	$\pm$	\ pm	$\prime$	\ prime
ϵ	\ epsilon	ρ	\ rho	Ξ	\ Xi	$\times$	\ times	$\Leftarrow$	\ Leftarrow
ζ	\ zeta	σ	\ sigma	Π	\ Pi	∞	\ infty	$\angle$	\ angle
η	\ eta	τ	\ tau	Σ	\ Sigma	$\sum$	\ sum	$\surd$	\ surd
θ	\ theta	υ	\ upsilon	Υ	\ Upsilon	$\int$	\ int	#	\ #
ι	\ iota	ϕ	\ phi	Φ	\ Phi	$\div$	\ div	$	\ $
κ	\ kappa	χ	\ chi	Ψ	\ Psi	$\sim$	\ sim	%	\ %
λ	\ lambda	ψ	\ psi	Ω	\ Omega	$\leftarrow$	\ leftarrow	&	\ &
μ	\ mu	ω	\ omega			$\uparrow$	\ uparrow	$\mid$	\ |

数学符号通过在其特殊拼音前加反斜杠(\ )来生成。一些常用符号也在表 6.1 中给出。常用句法是在一对单引号间放置一组相关指令。当一组符号要放置在一起时，如在指数中显示的表达式，则可将它们放在一对大括号({})之内。下面举例说明这些过程。

当 $\beta=3, 1\leq\Omega_1\leq 2$ 时，计算以下函数并对相应图形进行标注：

① 参见 L.Lamport, *LaTeX: A Document Preparation System*, Addison-Wesley, Reading, MA, 1987 年。

$$g_2 = 1 + e^{-\Omega_1^\beta}$$

程序如下：

```
Omega1 = linspace(1,2);beta = 3;
plot(Omega1,1 + exp( - Omega1.^beta),'k')
title('Plot of g _ 2 versus \ Omega _ 1 for \ beta = 3')
ylabel('g _ 2')

xlabel(' \ Omega _ 1')
text(1.2,1.2,'g _ 2 = 1 + e^{ - \ Omega _ 1^ \ beta}','fontsize',16)
```

结果如图 6.14 所示。

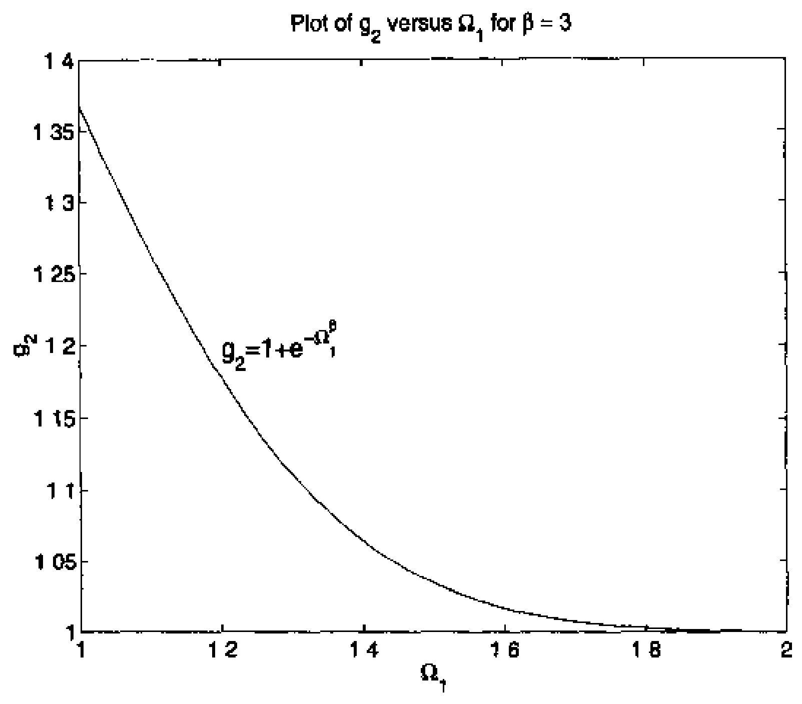


图 6.14 用上标、下标和希腊字母对图形进行标注

6.3.2 重复曲线：在 $0 \leqslant x \leqslant m\pi$ 范围内绘制 cot(x) 曲线

下面将在 $0 < x < m\pi$ 范围内显示 cot(x)，其中 m 由用户选择，可以是 2,3, …或 6。$m = 2$ 时的图形如图 6.15 所示。y 轴的上下限为 ±8，x 轴的取值范围从 0 到 $m\pi$。因为 $\cot(0) = \infty$，$\cot(\pi) = -\infty$，所以 x 的取值不包括 0 和 π。因为余切函数周期为 $m\pi$，所以只须画出 $0 < x < \pi$ 间的函数图形，然后将其右移 $(m-1)\pi$ 即可得到 $(m-1)\pi < x < m\pi$，$m > 1$ 区域内的函数图形。程序如下：

```
m = input('Enter number of repetitions of cot function (integer from 2 to 6) = ');
the = linspace(0.12,pi - 0.12,50);
ct = cot(the);
hold on
for n = 1:m
  plot([(2 * n - 1) * pi/2(2 * n - 1) * pi/2],[ - 8/40 8/40],'k - ')
  if n = = 1
text((2 * n - 1) * pi/2 - pi/(8 * m), .6,'/pi/2')
  text(n * pi + pi/(8 * m), - 7.5,' \ pi')
else
```

```
text((2*n-1)*pi/2-pi/(8*m),0.6,[num2str((2*n-1),2)'\pi/2'])
  if n<m
    text(n*pi+pi/(8*m),-7.5,[num2str(n,1),'\pi'])
  end
end
if n==m
  plot(the+(n-1)*pi,ct,'k-')
else
  plot(the+(n-1)*pi,ct,'k-',[n*pi n*pi],[-8 8],'k--')
  end
end
plot([0 m*pi],[0 0],'k-')
axis([0 m*pi -8 8])
xlabel('x (radians)')
ylabel('cot(x)')
title(['Cotangent function from 0 to 'num2str(m,1),'\pi'])
box on
```

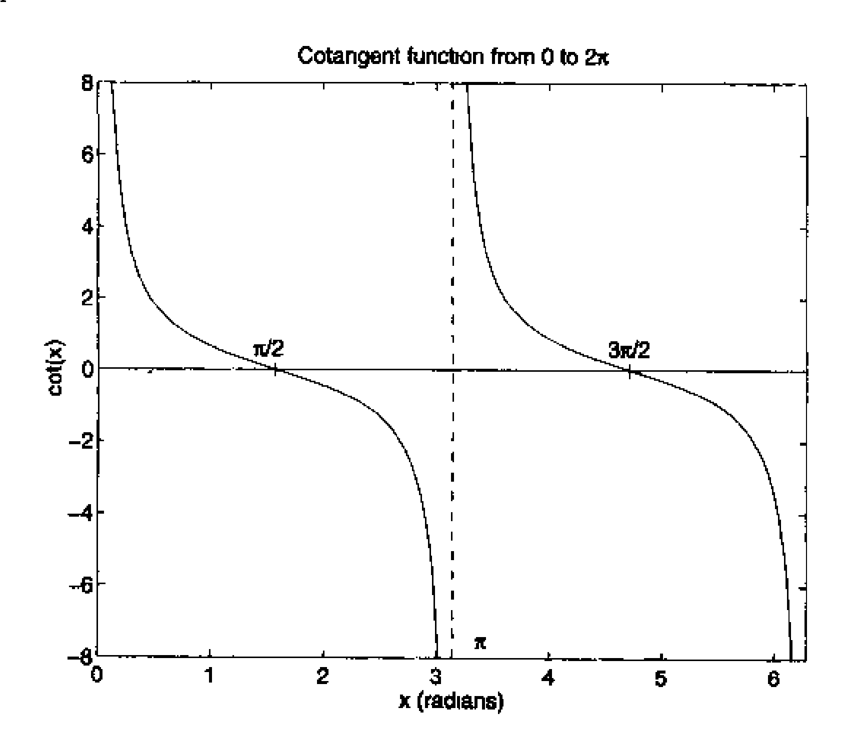


图 6.15 0 到 2π 区间内的余切函数

几乎可以看出所有的程序行都用于函数图形的标注，仅有两行语句用于绘制 cot(x)的函数图形。

6.3.3 极坐标图形：声源的远域辐射模式

无限大挡板内的圆形活塞以频率 f 振动，距离其中心极远处的标准声压由下式给出：

$$p(r,\theta)=\left|\frac{J_1(ka\theta)}{ka\theta}\right| \quad ka^2<<r \quad \text{和} \quad a<<r$$

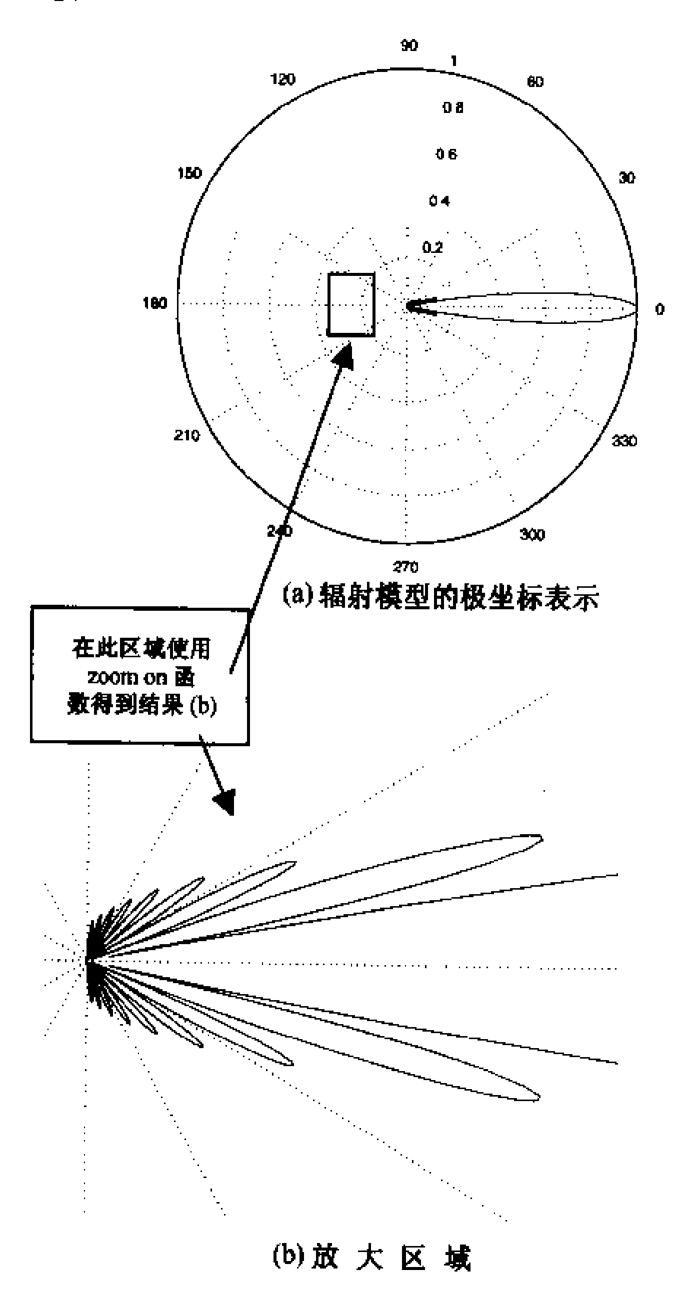
其中 r 为距离活塞圆心的极半径，θ 为 r 与挡板平面之间的角度，k 为波数，a 为活塞半径，$J_1(x)$为一类 1 阶贝塞尔函数。波数是频率为 f 的声波波长的倒数，因此 ka 是无量纲的。该模型是对扩音器声波角扩散的一个很好的近似。

创建一个标准辐射模型的极坐标，其中 $ka=6\pi$，$-\pi/2<\theta<\pi/2$。该例同时也可说明

zoom 函数的用法，正如下面所看到的，在 MATLAB 的极坐标图中，该辐射模型所展示的侧叶瓣不够清晰。程序如下：

```
theta = linspace( - pi/2,pi/2,300);
rad = abs(besselj(1,6 * pi * theta)./(6 * pi * theta));
polar(theta,rad/max(rad))
zoom on
```

上述程序结果如图 6.16(a)所示。注意 θ 的值使 $ka\theta \neq 0$。max 函数得到向量 ***rad*** 的最大值，因此 *rad*/max(*rad*)的比值是标准的辐射模型，其最大值为 1。zoom on 函数允许用户用鼠标左键定义一个矩形区域，并在释放鼠标键时使其填充绘图空间。矩形区域如图 6.16(a)所示，释放鼠标键后，其图形如图 6.16(b)所示。为使图形恢复成原始大小，当光标在图形中时可按鼠标右键 1 到数次。可以通过键入 zoom 或 zoom off 使缩放特性关闭，也可以在图形窗口中通过缩放图标来完成上述过程。



(a) 辐射模型的极坐标表示

(b) 放 大 区 域

图 6.16　辐射模型的极坐标图

6.3.4 多个图形的绘制:周期脉冲序列和单脉冲的频谱图

如图 6.17(a)所示,一个周期矩形脉冲序列,脉宽为 d,周期为 T,该信号可表示为(见例 2.3):

$$g(t) = \frac{f(t)T}{Ad} = c_0 + 2\sum_{n=1}^{\infty} c_n \cos(n\omega_0 t)$$

其中 $\omega_0 = 2\pi/T$ 且,

$$c_0 = 1$$

$$c_n = \frac{\sin(n\pi d/T)}{(n\pi d/T)} \qquad n = 1,2,\cdots$$

c_n 为信号谐波的幅值,其模$|c_n|$为 n 的函数,在图中画出该函数,则称为信号的幅度谱。频谱图中只有 ω_0 处(即 $n\omega_0$ 处)谐波分量不为 0,其余频率的谐波分量均为 0。由下式中可看出,当 $c_n = 0$ 时,

$$\frac{n\pi d}{T} = m\pi \quad 或 \quad n = \frac{m}{(d/T)}$$

另一方面,如果有一单脉冲,如图 6.17(b)所示,它的频谱为:

$$F(0) = 1$$

$$F(\omega) = \frac{G(\omega)}{Ad} = \frac{\sin(\omega d/2)}{(\omega d/2)} \quad \omega > 0$$

$G(\omega)$的绝对值称为振幅密度频谱。

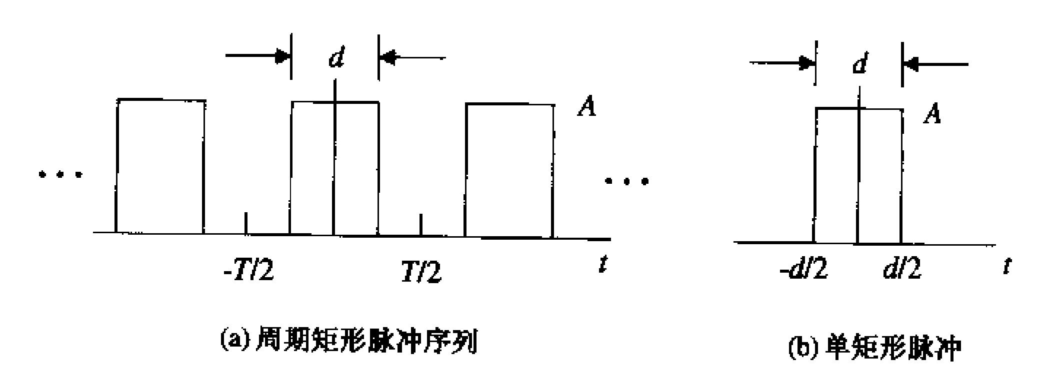
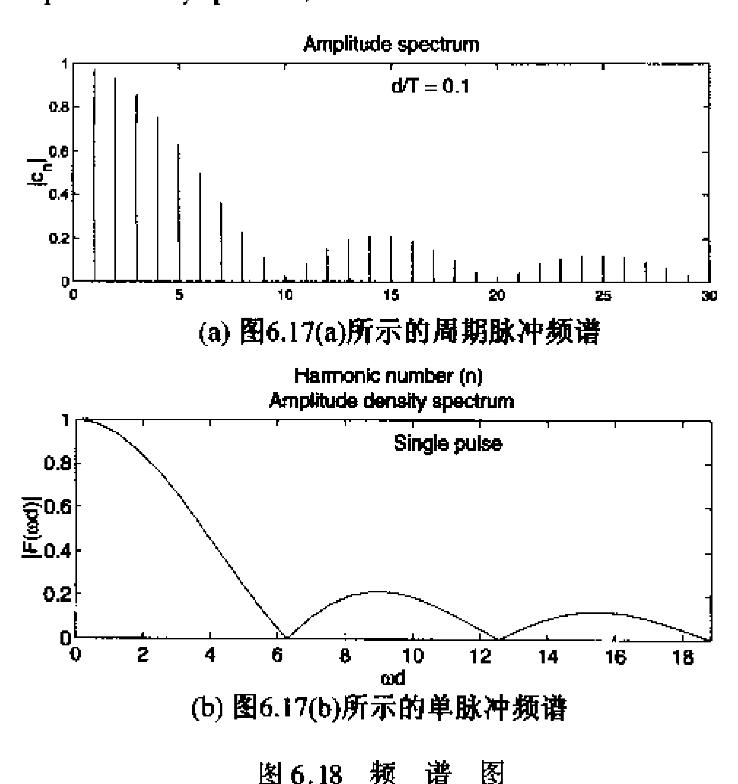


图 6.17 矩形脉冲

先创建两个图形:一个是$|c_n|$对 n 的变化图,其中 $n = 1,2,\cdots, 30$ 且 $d/T = 0.1$;第二个是$|F(\omega d)|$对 ωd 的变化图,其中 $0 \leqslant \omega d \leqslant 6\pi$。为创建第一个图形,使用第 6.2.2 节中的方法画一系列未连接的直线。画出的两个图形一个在另一个的上方。注意不要使除数为 0。下述程序结果如图 6.18 所示。

```
n = 1:30;
cn = [1 abs(sin(0.1 * pi * n)./(0.1 * pi * n))];
n = [0 n];
subplot(2,1,1)
plot([n;n],[zeros(1,length(cn));cn],'k')
xlabel('Harmonic number (n)')
ylabel('|c_n|')
```

```
text(15,.9,'d/T=0.1')
title('Amplitude spectrum')
w=pi/5:pi/5:6*pi;
subplot(2,1,2)
plot([0 w],[1 abs(sin(w/2)./(w/2))],'k')
axis([0 6*pi 0 1])
xlabel('\omegad')
ylabel('|F(\omegad)|')
text(3*pi,.9,'Single pulse')
title('Amplitude density spectrum')
```



(a) 图6.17(a)所示的周期脉冲频谱

(b) 图6.17(b)所示的单脉冲频谱

图6.18 频 谱 图

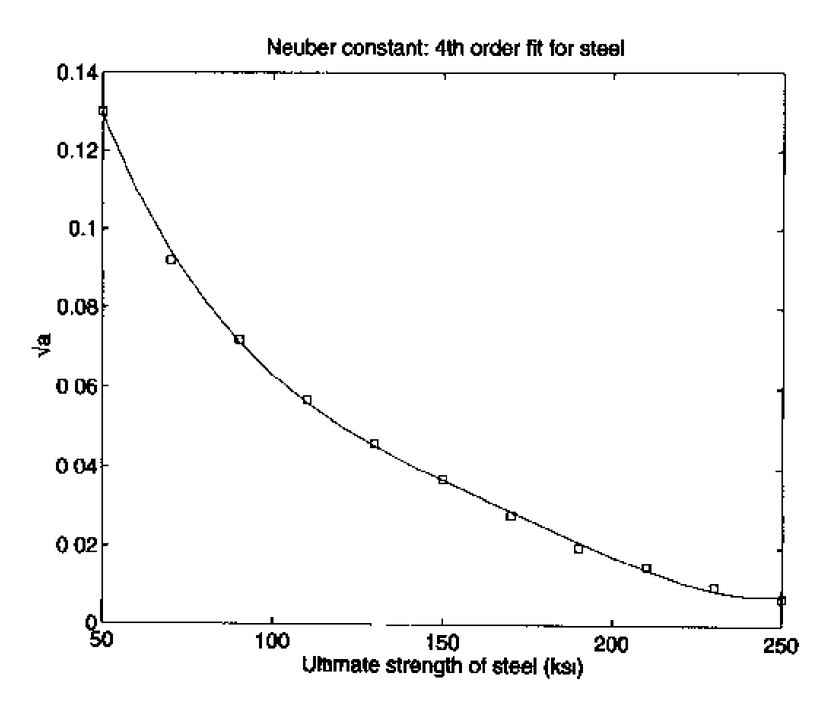
可采用如下方法避免除数为0的操作:谐波次数向量最初定义为 $n=1:30$,这样用点除即可方便地计算出在此范围内的 $|c_n|$ 值,然后分别创建包括 $n=0$ 和 $c_0=1$ 的两个向量,如第二条和第三条曲线所示。采用同样的方法可计算出 $|F(\omega)|$ 的值。该方法不需任何编程逻辑。

6.3.5 多条曲线的绘制:钢的槽口敏感性

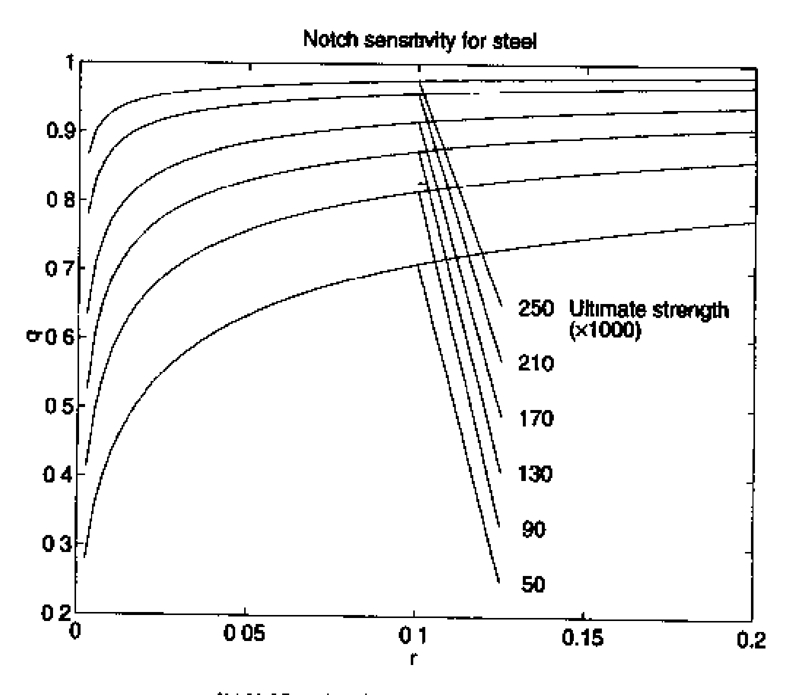
参考第5.6.1节的例子,在 $50\leqslant S_u\leqslant 250$ 和 $0<r<0.2$ 范围内画出槽口敏感性常数 q 的曲线。为提高程序可读性,对要填充的数据创建一个函数,如下所示:

```
function ns=DataNeuber
ns=[50,.13;70,.092;90,.072;110,.057;130,.046;150,.037;...
    170,.028;190,.020;210,.015;230,.010;250,.007];
```

下述程序由两部分组成。第一部分求得四次多项式的系数,并画出数据点及这些点处的多项式值。第二部分用多项式创建曲线族,这些曲线为钢的极限强度 S_u 取不同值时槽口敏感性常数 q 对槽口半径的关系曲线。程序执行结果如图6.19(a)和6.19(b)所示。



(a) 钢的纽博尔常数



(b) 钢的槽口敏感性对槽口半径 r 的函数曲线

图 6.19 钢的极限强度

```
st = 50:10:250;skip = 1:2:11;loc = 0.25:0.08:0.65;
ncs = DataNeuber;
p = polyfit(ncs(:,1),ncs(:,2),4);
figure(1)
plot(st,polyval(p,st),'k',ncs(:,1),ncs(:,2),'ks')
title('Neuber constant:4th order fit for steel')
xlabel('Ultimate strength of steel (ksi)')
ylabel('\surda')
figure(2)
```

```
[s,r]=meshgrid(ncs(skip,1),0.0025:0.0025:0.2);
notch=inline('1./(1+polyval(p,s)./sqrt(r))','p','s','r'r)
plot(r,notch(p,s,r),'k')
hold on
plot([repmat(0.125,1,6);repmat(0.1,1,6)],[loc;notch(p,ncs(skip,1)',0.1)],'k')
text(repmat(0.013,1,6),loc,num2str(ncs(skip,1)))
text(0.145,0.65,'ultimate strength')
text(0.145,0.62,'( \ times 1000)')
xlabel('r')
ylabel('q')
title('Notch sensitivity for steel')
```

meshgrid 函数生成了两个(80×6)维数组,其中行是 r 的值,列是 S_u 的值。由于 inline 函数中的表达式 *notch* 使用点标记,所以可以在数组中得到合适的参数。最后一个 plot 函数放置曲线标识线,如前所述,画出一系列未连接的直线。但必须将列向量 ***ncs***(:,1)转换为行向量。为了在线段末端加上文字注释,可创建 x,y 的坐标向量,其相应文字由 num2str 函数将 ***ncs***(:,1)的第 1 列转换为字符串而得到。将其第 1 个下标加 2 可选择 ***ncs*** 的另一元素。

6.3.6 不同 *y* 轴的多曲线绘制:plotyy

两个无量纲量分别是压力比和温度比,每一个都是高度比 h/h_L 的函数。比值由下式给出:

$$\frac{T}{T_o}=\left(1+\frac{h}{h_L}\right)^{1.5} \quad 和 \quad \frac{P}{P_0}=e^{-h/h_L}$$

下列程序说明了在同一标注图中如何用 plotyy 函数在 $0 \leqslant h/h_L \leqslant 1$ 的范围内画出两条比率曲线的方法。程序执行结果如图 6.20 所示。

```
hoverhL=linspace(0,1,10);
ToverTo=(1+hoverhL).^1.5;
PoverPo=exp(-hoverhL);
[ax,h1,h2]=plotyy(hoverhL,PoverPo,hoverhL,ToverTo);
xlabel('h/h_L')
ylabel('P/P_o')
v=axis;
text(v(2)*1.06,v(3)+(v(4)-v(3))/2,'T/T_o','rotation,'90)
text(v(1)+(v(2)-v(1))/5,v(3)+(v(4)-v(3))/1.6,'Pressure')
text(v(2)/1.6,v(4)/1.2,'Temperature')
set(h2,'Marker','s')
set(h1,'Marker','<')
```

函数 plotyy 画出每条曲线并给出了正确的标度值。它把左边的纵轴值赋给了 plotyy 函数参数中的第一对变量,右边的纵轴值给第二对变量。但只能用 ylabel 标示左边的纵轴,且 plotyy 函数不支持对每条曲线线型的定义。因此,为标注右边的纵轴并给出每条曲线自己的线型定义,使用在程序中给出的 plotyy 的形式。$ax(1)$和 $ax(2)$分别为左侧轴和右侧轴的句柄。$h1$ 和 $h2$ 分别为第一条和第二条曲线的句柄,其顺序由 plotyy 函数确定。set 函数用于设置其第 1 个参数所表示的句柄的 Marker 属性。第一个 set 函数设定用正方形画每个点,第二个 *set* 函数则设定为用指向左侧的三角形画点。第一个 text 语句中的属性 rotation 使文本旋转了 90°,如图

中所示。

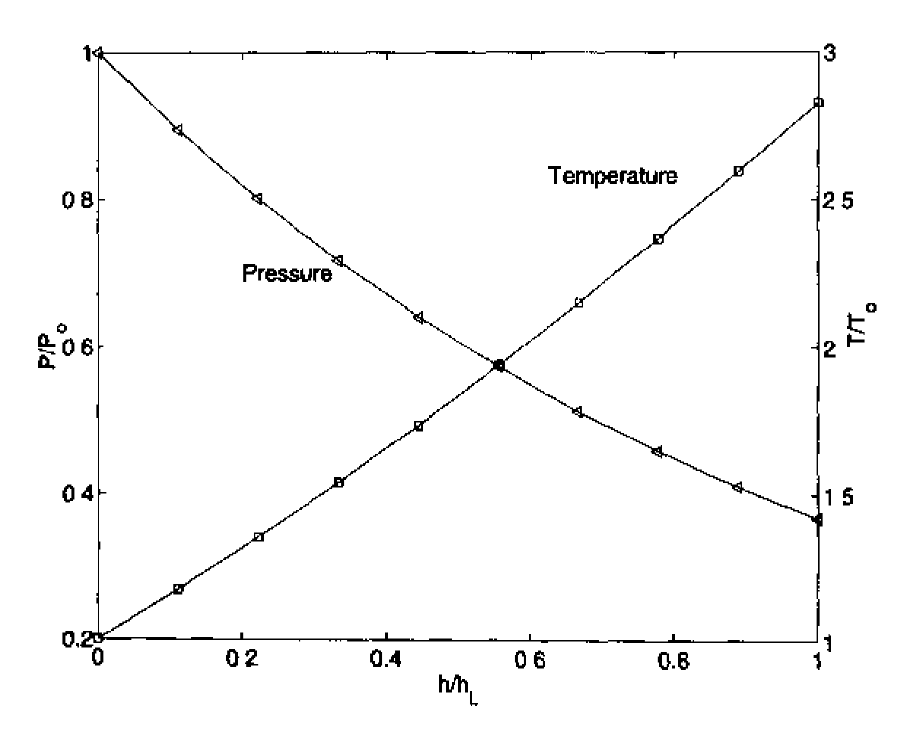


图 6.20 由 plotyy 绘制的两个不同参数

6.3.7 从图形读取数值:ginput

在 MATLAB 中,可以用 ginput 函数直接从图形中获取 (x, y) 坐标值。使用时,可以在 MATLAB 命令窗口中直接输入该函数,也可以在程序中引用。下面用第 5.6.3 节所创建的 *DampSineWave* 函数编写一段程序以说明 ginput 函数的使用方法。目的是从图形中得出衰减周期,其方法为:读出几个连续最大振幅处的对应值并求其平均值,在函数的过零点处读取几个连续值并求其平均值,将这些平均值与下述分析得到的周期 T 相比较:

$$T = \frac{2\pi}{\sqrt{1-\xi^2}}$$

其中 ξ 是无量纲的阻尼系数。

该函数的典型应用形式为:

```
[x,y] = ginput
```

其中 x 和 y 分别为一个向量值,对应在图形中鼠标左键按下处光标(十字中心)的坐标值。光标可多次定位。最后一个点记录后,单击 Enter 键确认。采用该方法时,为得到最佳的结果,记录光标坐标值以前应使图形窗口最大化。

令 $\xi = 0.1, 0 \leqslant \tau \leqslant 30$,程序如下:

```
tau = linspace(0,30,200);xi = 0.10;
plot(tau,DampedSineWave(tau,xi),'k')
grid on
[tim,amp] = ginput;
disp(' tau   ampl')
disp(num2str([tim amp]))
disp([Analytically determined period = 'num2str(2 * pi/sqrt(1 - xi^2))])
```

```
disp(['Average period graphically obtained = 'num2str(mean(diff(tim)))])
disp(['Standard deviation = 'num2str(std(diff(tim)))])
```

执行程序，由图中选定过零点处的5个值，则得到如下结果：

```
tau            ampl
1.68539        -0.00137931
7.99358        -0.00137931
14.3018        -0.00137931
20.61          -0.00137931
26.9181        -0.00137931
Analytically determined period = 6.3148
Average period graphically obtained = 6.3082
Standard deviation = 2.2933e-0.15
```

如在最大值处选定5个点，结果如下：

```
tau          ampl
0            0.995862
6.30819      0.528276
12.5682      0.28
18.9246      0.147586
25.2327      0.0772414
Analytically determined period = 6.3148
Average period graphically obtained = 6.3082
Standard deviation = 0.039318
```

6.3.8 使用随机数填充区域

可以使用随机产生的坐标值来决定一个三角形的近似面积，仅用位于三角形区域内的点来填充该三角形并显示。随机数由下述语句产生：

```
r = unifrnd(a1,a2,m,n)
```

该语句生成一个($n \times m$)的数组，且数组中的任意值落在(a_1,a_2)区间内的概率相同。

如图6.21中所示三角形，其面积 $A_e = 0.5bc$。三角形近似面积是分布于边界内随机点数与总面积 bc 的乘积。若令 $b=1, c=2, a=c/4$，考虑3000对点的情况，程序如下：

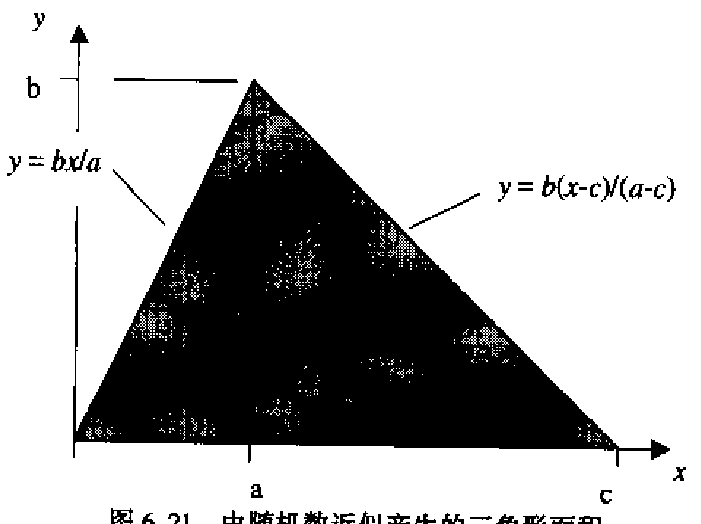


图6.21 由随机数近似产生的三角形面积

```
b=1;c=2;a=0.25*c;
exactarea=0.5*b*c;
N=3000;
x=unifrnd(0,c,1,N);
y=unifrnd(0,b,1,N);
indL=find(x<=a);
nxL=x(indL);
nyL=y(indL);
indyL=find(nyL<=b*nxL/a);
indR=find(x>a&x<=c);
nxR=x(indR);
nyR=y(indR);
indyR=find(nyR<=b/(a-c)*(nxR-c));
approxA=b*c*(length(indyL)+length(indyR))/N;
plot(nxL(indyL),nyL(indyL),'k.',nxR(indyR),nyR(indyR),'k')
text(0.6*c,0.95,['N='num2str(N)])
text(0.6*c,0.90,['Approx area='num2str(approxA)])
text(0.6*c,0.85,['Exact area='num2str(exactarea)])
```

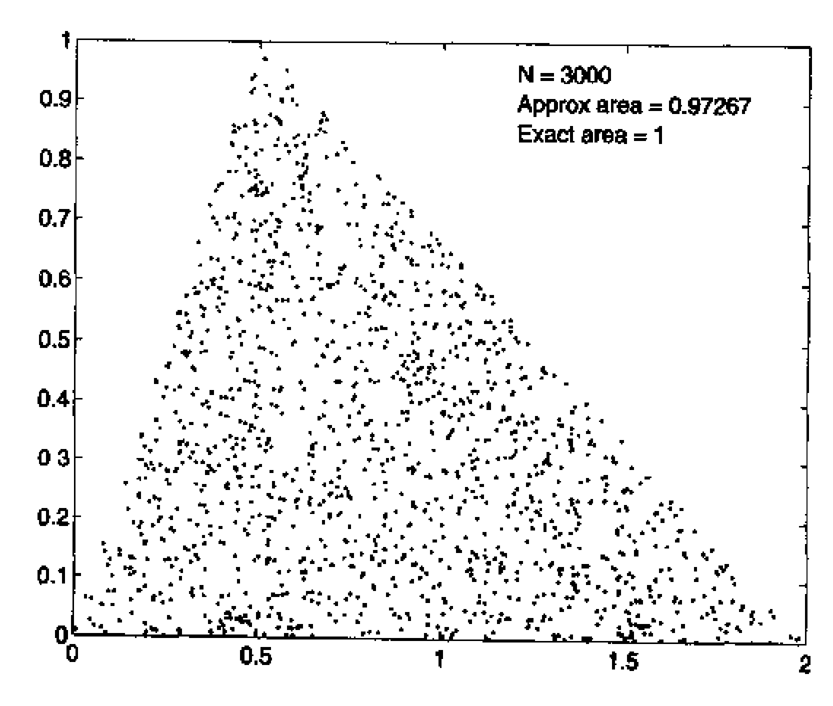


图 6.22 由随机点填充的三角形

程序执行结果如图 6.22 所示。

练习

注意：练习中图形的绘制可通过向量、点运算和 meshgrid 函数实现。for 循环仅用于提高参数的适用范围。

6.1 施加在贝尔维尔(Belleville)弹簧(参见图 5.33)上的力与 C_1 成比例，其中，

$$C_1 = 0.5d_t^3 - 1.5h_t d_t^2 + (1 + h_t^2)d_t$$

$h_t = h/t$，$d_t = \delta/t$，δ 为弹簧的挠度。当 h_t 在 1 到 3 之间变化，增量为 0.25 且 d_t 在

0到5之间变化时，画出 C_1 随 d_t 变化的曲线，标注曲线并令 y 轴极限值为8。结果应如图6.23所示。

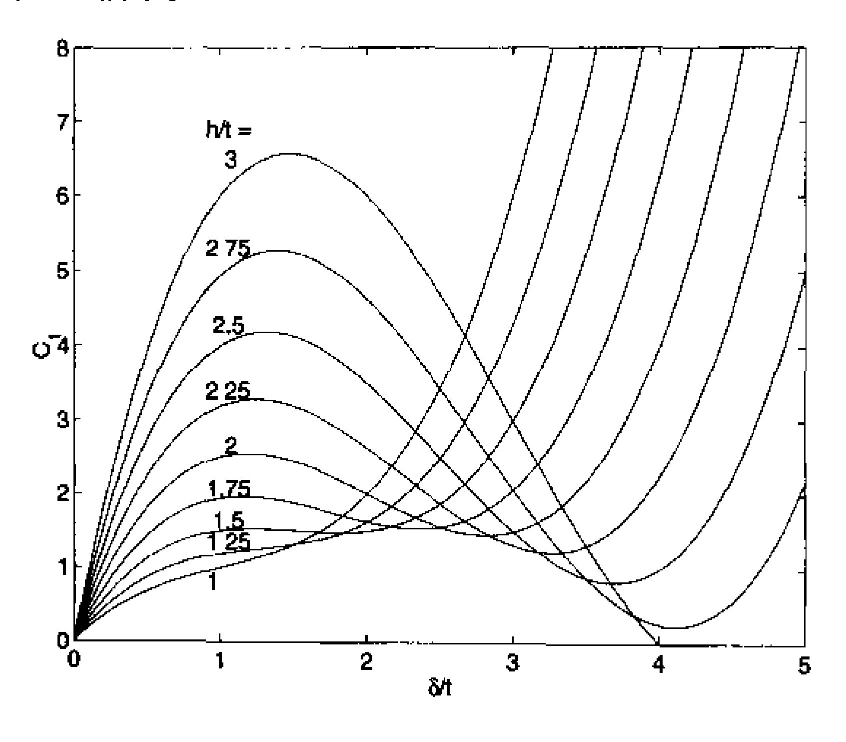


图6.23　贝尔维尔弹簧的系数 C_1

6.2 如图6.24所示齿轮齿，如果齿轮有 n 个齿，则每个齿占 $2\pi/n$ 度。设 R_b 为基圆半径，R_T 为齿节圆的半径，$R(R_b \leq R \leq R_T)$ 为齿轮廓上某点的半径。则轮齿轮廓极坐标为 (R, Ψ)，包括相邻齿间的空隙，如表6.2所示。表6.2中，φ_s 为齿轮压力角，值为14.5°、20°或25°，$R_s = nm/2$ 为标准齿节半径，m 为齿轮模数，t_s 为 R_s 处的齿轮厚度。

表6.2　图6.24所示齿轮齿的不同部位的定义

R	Ψ	定义
$R_b \leq R \leq R_T$	$\text{inv}(\varphi(R))$	$\varphi(R) = \arccos(R_b/R)$
		$\text{inv}(x) = \tan(x) - x$
R_T	$\text{inv}(\varphi(R_T)) \leq \Psi \leq \text{inv}(\varphi(R_T)) + 2\theta_T$	$\theta_T = 0.5t_s/R_s + \text{inv}(\varphi_s) - \text{inv}(\varphi(R_T))$
$R_b \leq R \leq R_T$	$2[\theta_T + \text{inv}(\varphi(R_T))] - \text{inv}(\varphi(R))$	
R_b	$2[\theta_T + \text{inv}(\varphi(R_T))] \leq \Psi \leq 2\pi/n$	

如果齿轮有24齿，压力角为20°，模数为10 mm，齿厚为14.022 mm，基圆半径为90.21 mm，齿节圆半径为106 mm，用两种方式：plot和polar画出齿轮图形。

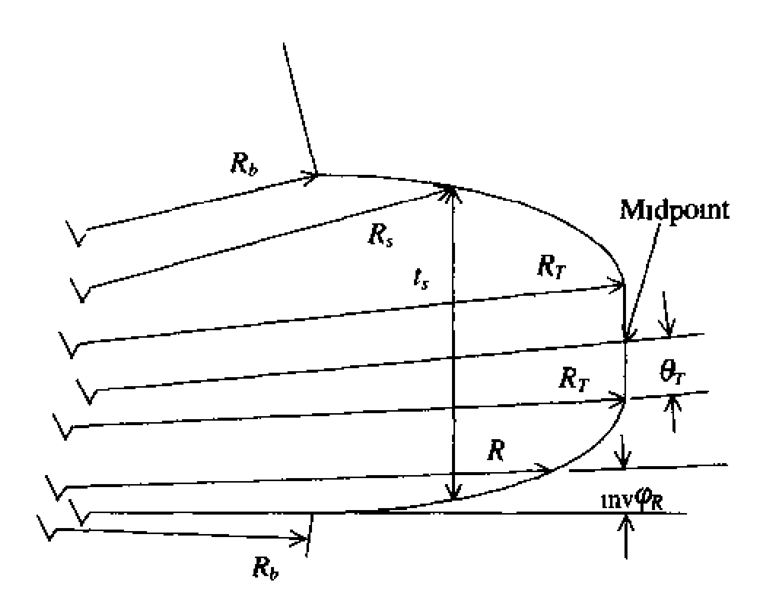


图 6.24 齿轮齿的几个参数

6.3 忽略轴环的摩擦，电钻效率的百分比为(参见练习 8.8)：

$$e = 100\,\frac{\cos(\alpha) - \mu\tan(\lambda)}{\cos(\alpha) + \mu\cot(\lambda)}\%$$

其中 μ 为摩擦系数，λ 为电钻的超前角，α 为螺纹角。当 $0<\lambda<90°$，$\mu=0.02,0.05,0.10,0.15,0.20,0.25$，螺纹角 $\alpha=7°$和 $14.5°$时画出效率随 λ 变化的函数曲线、标注图形及各条曲线，并用 axis 函数限制效率范围为 0% 到 100%。结果应如图 6.25 所示。

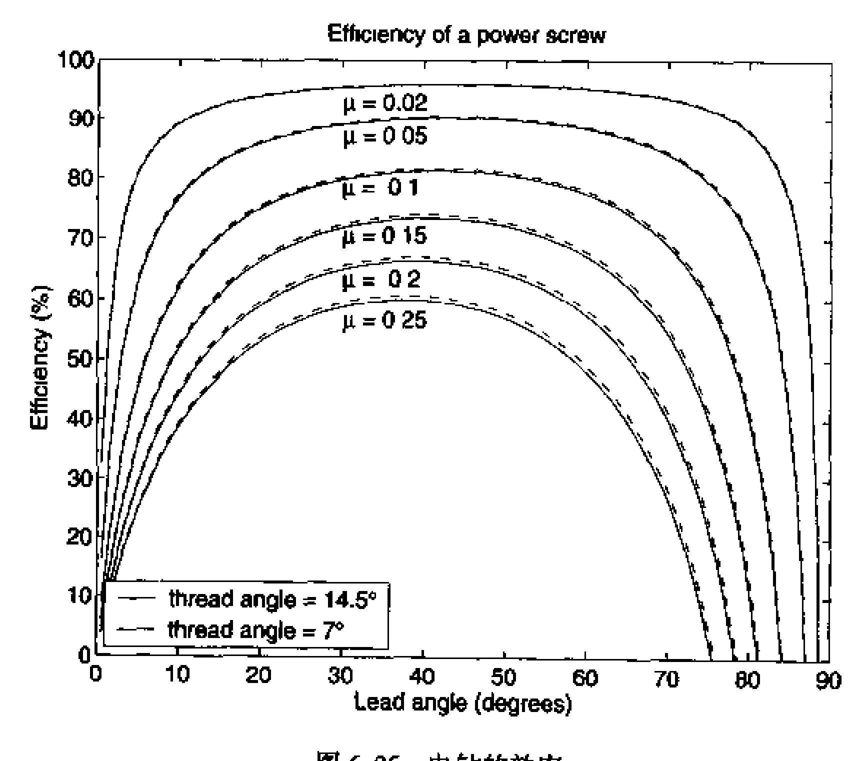


图 6.25 电钻的效率

6.4 使用练习 1.1 的结果，画出 $v_1=0.3$ 时，σ_x/p_{max}，σ_z/p_{max}，$\tau_{xz}/p_{max}=\tau_{yz}/p_{max}=0.5(\sigma_x/p_{max}-\sigma_z/p_{max})$随 z/a 变化的图形。对图形和曲线加以标注。

6.5 使用练习 1.2 的结果,画出 $v=0.3$ 时,σ_x/p_{max},σ_y/p_{max},σ_z/p_{max},τ_{yz}/p_{max}随 z/b 变化的图形。对图形和曲线加以标注。

6.6 画出下面的曲线,[①] 要求使用 axis equal 函数。

摆线($-\pi\leqslant\varphi\leqslant3\pi$; $r_a=0.5,1,1.5$):

$$x=r_a\varphi-\sin\varphi$$
$$y=r_a-\cos\varphi$$

双纽线($-\pi/4\leqslant\varphi\leqslant\pi/4$):

$$x=\cos\varphi\sqrt{2\cos(2\varphi)}$$
$$y=\sin\varphi\sqrt{2\cos(2\varphi)}$$

螺旋线($0\leqslant\varphi\leqslant6\pi$):

阿基米德曲线:

$$x=\varphi\cos\varphi$$
$$y=\varphi\sin\varphi$$

对数曲线($k=0.1$):

$$x=e^{k\varphi}\cos\varphi$$
$$y=e^{k\varphi}\sin\varphi$$

心形线($0\leqslant\varphi\leqslant2\pi$):

$$y=2\cos\varphi-\cos2\varphi$$
$$y=2\sin\varphi-\sin2\varphi$$

星形线($0\leqslant\varphi\leqslant2\pi$):

$$x=4\cos^3\varphi$$
$$y=4\sin^3\varphi$$

外摆线($R_r=3$, $a_r=0.5,1$ 或 2, $0\leqslant\varphi\leqslant2\pi$; $R_r=2.5$, $a_r=2$ 且 $0\leqslant\varphi\leqslant6\pi$):

$$x=(R_r+1)\cos\varphi-a_r\cos(\varphi(R_r+1))$$
$$y=(R_r+1)\sin\varphi-a_r\sin(\varphi(R_r+1))$$

内摆线($R_r=3$, $a_r=0.5,1$ 或 2, $0\leqslant\varphi\leqslant2\pi$):

$$x=(R_r-1)\cos\varphi+a_r\cos(\varphi(R_r-1))$$
$$y=(R_r-1)\sin\varphi-a_r\sin(\varphi(R_r-1))$$

6.7 油的绝对粘度可由下式估算,单位为 μreyn (lb-s/in^2),其误差为 ±10%:

$$\mu=10^{C-1}$$

其中,

$$C=10^{A_j-B_j\log_{10}T_o}$$

$T_o=255.2+5/9T$ °K,其中 T 为油温,单位为°F,A_j 和 B_j 为油 SAE(简单算术表达式)数的函数,其值由表 6.3 给出。画出表 6.3 中六种油的$\log_{10}\log_{10}(10\mu)$随 $\log_{10}(T_0)$变化的函数关系曲线及 μ 随 T_0 变化的函数曲线(参见第 8.6 节)。

① 参见 D von Seggern, *CRC Standard Curves and Surfaces*, CRC Press, Inc , Boca Raton, FL, 1993 年。

表 6.3　确定 μ 的系数

SAE 数	j	A_j	B_j
10	1	9.1209	3.5605
20	2	9.1067	3.5385
30	3	8.9939	3.4777
40	4	8.9133	3.4292
50	5	8.5194	3.2621
60	6	8.3666	3.1884

6.8 蜗轮的超前角为 λ；齿数比 $\beta = N_1/N_2$，其中 N_1 是蜗轮的齿数，N_2 是驱动轮的齿数；轴间中心距 C 和法向径节 P_{dn} 之间的关系为：

$$K = \frac{2P_{dn}C}{N_2} = \frac{\beta}{\sin\lambda} + \frac{1}{\cos\lambda}$$

当 $1° \leqslant \lambda \leqslant 40°$，$\beta = 0.02, 0.05, 0.08, 0.11, 0.15, 0.18, 0.23, 0.30$ 时，画出 K 相对于 λ 的关系曲线。标注图形和曲线，y 轴范围限定在从 1 到 2 之间。在同一图形中用一条曲线连接每条曲线的最小值，从而得到练习 5.9(a)的结果。修改练习 5.9(a)中相关函数及部分程序，得到本题的程序。程序执行结果应如图 6.26 所示。

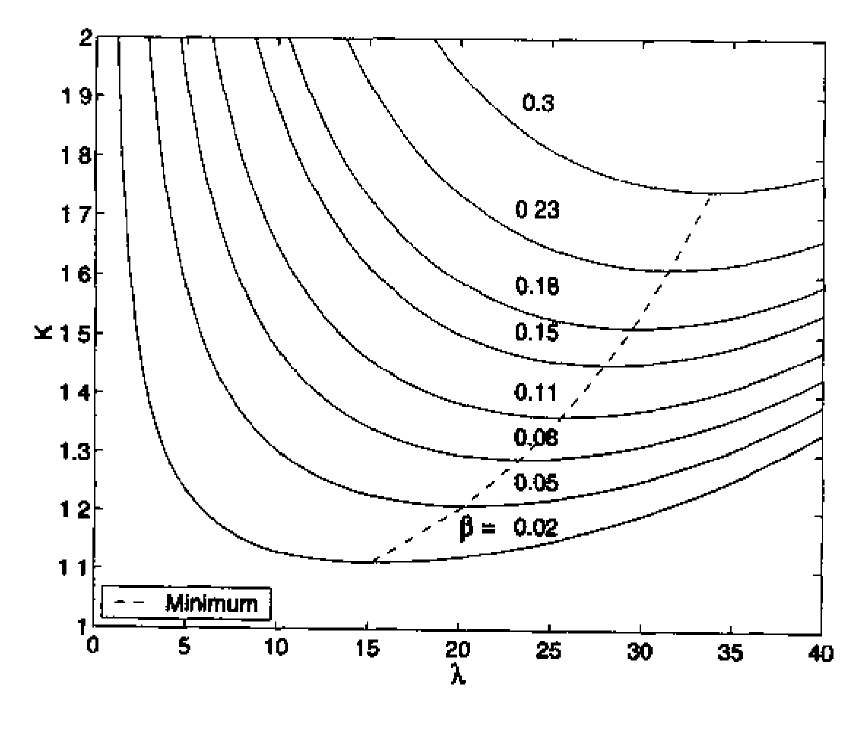


图 6.26　蜗轮的超前角

6.9 如图 6.27 所示，编制一段程序，使其在一个半径 $r_b = 1.5$ 的中心圆周围画出 3 个或 3 个以上的圆(图中是 5 个)。外部圆的半径 r_s 为：

$$r_s = \frac{r_b \sin(\pi/n)}{1 - \sin(\pi/n)}$$

程序要求用户输入需要画出的圆的个数，程序中未使用 for 循环。

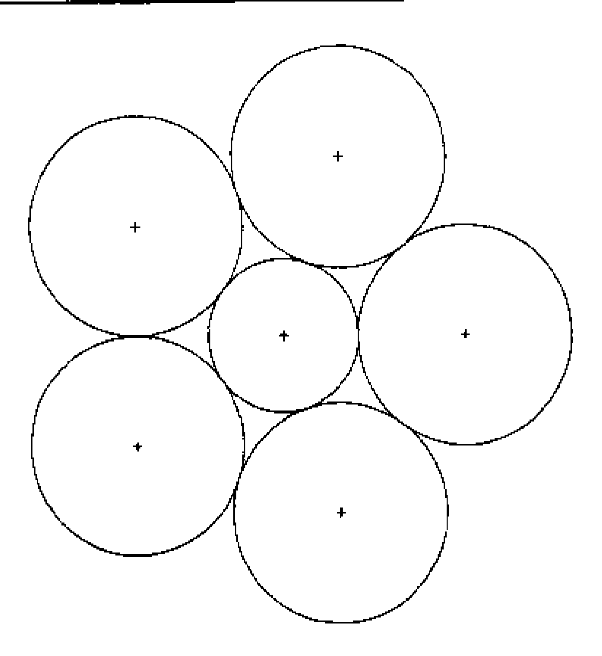


图6.27 一个圆周围的五个圆

6.10 在优化分析中画出被优化的函数(称为目标函数)和它的约束条件(要求解值限定在给定区域内)是一种好的方法。考虑下述最小化问题:

$$f(x_1,x_2)=(x_1-1)^2+(x_2-1)^2$$

其约束条件为:

$$g_1=(x_1-3)^2+(x_2-1)^2-1\leqslant 0$$

$$g_2=2x_1-x_2-5\leqslant 0$$

因此,解 x_{m1}和 x_{m2}一定在圆 $f(x_1,x_2)$上,并在 g_1,g_2 限定的区域内。

画出上述目标函数(圆圈)和解区间,结果应如图 6.28 所示。为此,应正确使用 fill 函数。

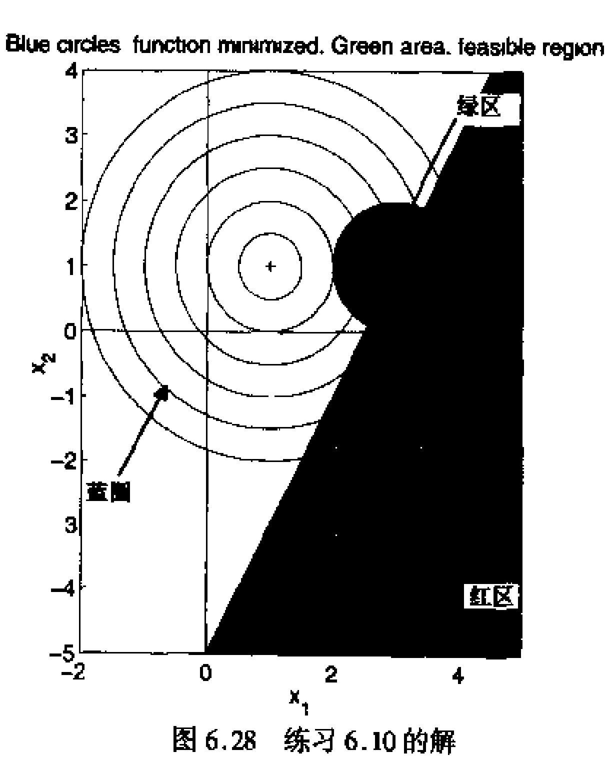


图6.28 练习6.10的解

6.11 练习 5.5 中给出的阔尔布鲁克通用公式如下，它用于估算管子的摩擦系数 λ：

$$\lambda = \left[-2\log_{10}\left(\frac{2.51}{R_e\sqrt{\lambda}} + \frac{0.27}{d/k}\right)\right]^{-2} \quad R_e \geqslant 4\,000$$

其中 R_e 为雷诺数，d 为管子的直径，k 为表面粗糙度。对于光滑管（$k \approx 0$；$d/k > 100\,000$）：

$$\lambda = \left[2\log_{10}\left(\frac{R_e\sqrt{\lambda}}{2.51}\right)\right]^{-2} \quad R_e \geqslant 4\,000$$

当 $4\times10^3 \leqslant R_e \leqslant 10^7$，$d/k$ = 20，50，100，200，500，1 000，2 000，5 000，10 000，20 000，50 000，100 000 和 ∞（$k=0$）时，画出 $\log_{10}(\lambda)$ 随 $\log_{10}(Re)$ 变化的曲线。使用 semilogx 函数代替 plot 函数。标注图形和曲线。在 $R_e = 10^7$ 的右侧放置各曲线标识符（即在图形右侧纵轴外侧）。结果即为著名的管道流动摩擦系数的莫迪图。结果应如图 6.29 所示。

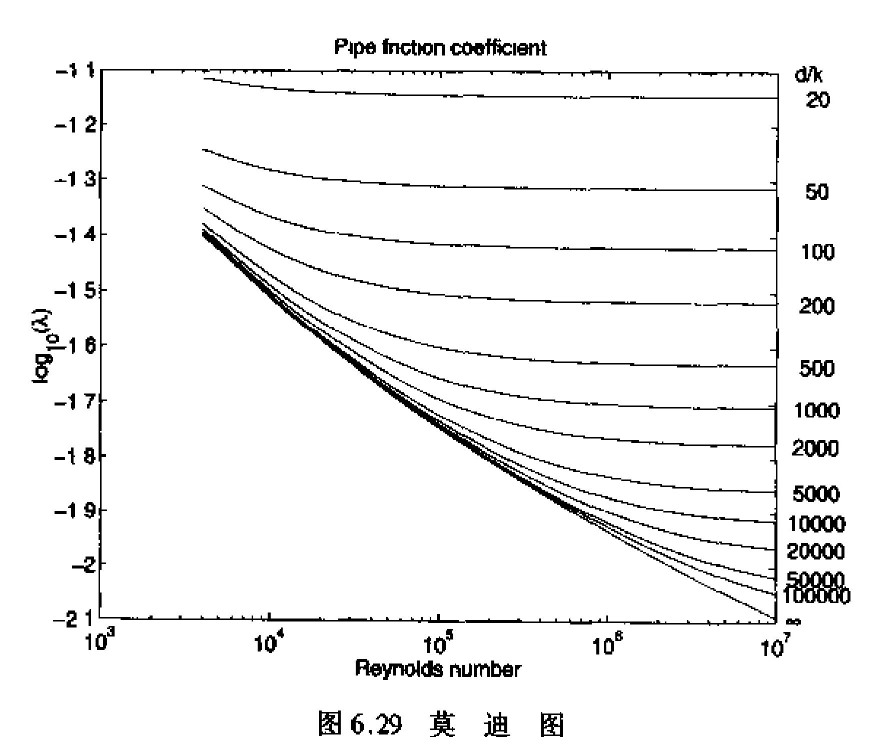
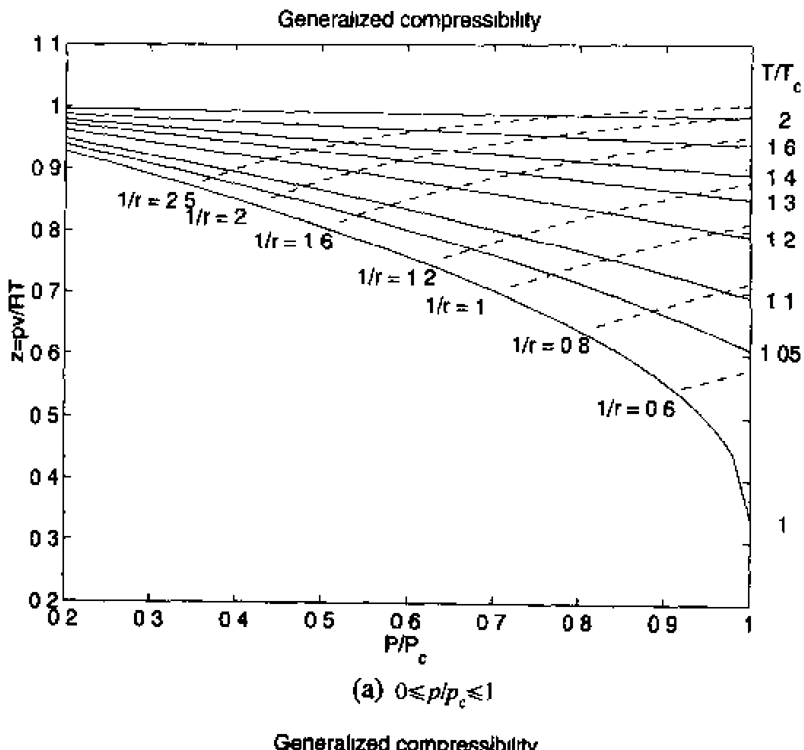


图 6.29 莫 迪 图

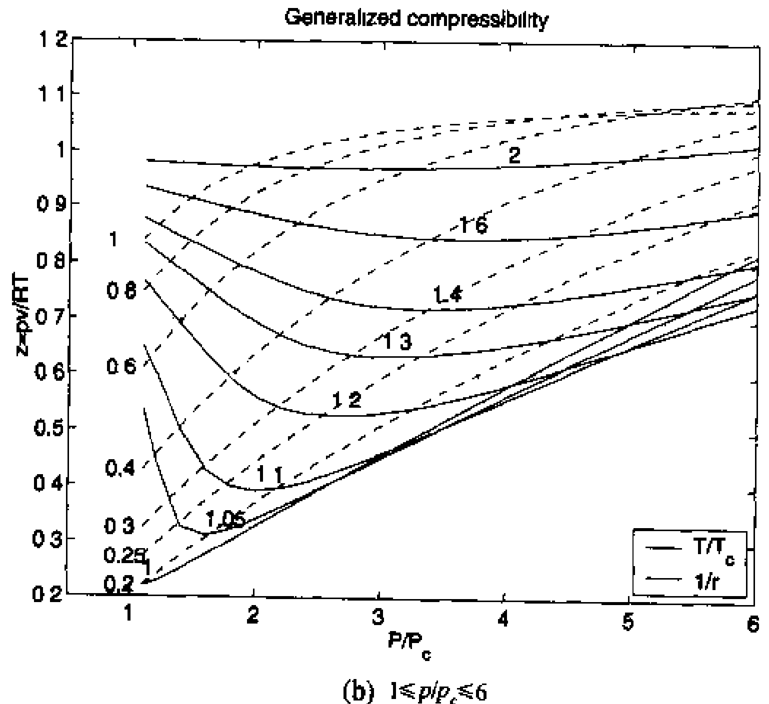
6.12 参见练习 5.4，其中用到的气体压缩方程如下：

$$Z(r,\tau) = \frac{p\tau}{r} = \frac{Pv}{RT}$$

其中 $Z(r,\tau)$ 在练习 5.4 中定义。使用练习 5.4(b)和练习 5.4(c)中所示的方法生成图 6.30(a)和图 6.30(b)。图 6.30(a)中，在函数 fzero 中，当 τ 是常数时，令 r 的取值范围为 $0.5 \leqslant r \leqslant 3.4$；当 r 是常数时，令 τ 的取值范围为 $0.4 \leqslant \tau \leqslant 2.5$。为了用虚线绘制曲线，使用 find 函数简化之。图 6.30(b)中，当 τ 是常数且小于 1/1.3 时，在 fzero 中使用假想值 $r=4$；当 $\tau \geqslant 1/1.3$，$p<3$ 时使用假想值 2；否则使用假想值 3.3。当 r 为常数时，使用假想值 $\tau = 0.6$。



(a) $0\leqslant p/p_c\leqslant 1$



(b) $1\leqslant p/p_c\leqslant 6$

图6.30 归纳出的可压缩性图

6.13 在 $-12\leqslant x\leqslant 7$ 范围内，存在如下多项式：

$$y=0.001x^5+0.01x^4+0.2x^3+x^2+4x-5$$

只画出正值。强迫 y 值在每段的始点和终点等于或接近于0。[提示：用 find 函数。]

6.14 编写一段程序创建练习5.6的图5.14。

6.15 如图6.31所示，一个 N 对声源的直线排列，声源的振动频率为 ω，振幅为 Q。排列中距离 r 处的总声压为：

$$P(r,\theta) = Z_o\left[\sum_{m=1}^{N}\left(1+\frac{md}{r}\cos\theta\right)\exp[j(-\varphi_m+mdk\cos\theta)]\right]$$

$$+\sum_{m=1}^{N}\left(1-\frac{md}{r}\cos\theta\right)\exp\left[j(-\varphi_{-m}-mdk\cos\theta)\right]$$

其中，

$$Z_o=\frac{j\rho ckQ}{4\pi r}e^{j(\omega t-kr)}$$

ρ 为介质密度，c 为在该介质中的音速，$k=2\pi\omega/c$ 为声波数，$\varphi_{\pm m}$ 为相对于 $\varphi 1=0$，$d/r<<1$，$dk<<1$ 的相位角。

本例中 $N=1$，$\varphi_1=0$，$\varphi_{-1}=\pi$，则上式可简化为：

$$P(r,\theta)=2Z_o\left[jdk+\frac{d}{r}\right]\cos\theta$$

该式为声音偶极子远域压力的表达式。

计算作为 θ 的函数 $|P(r,\theta)/Z_0|$，并当 $d/r=0.05$，$dk=0.03$，$\varphi_{-1}=\pi$，$\varphi_1=0$ 时，用 polar 函数画出 $0\leqslant\theta\leqslant 2\pi$ 时的结果，如图 6.32 所示。用此公式分析直线排列的一般情况，同时画出偶极子精确解与数值分析解的相对误差。该图中不包含第一点（$\theta=0$）。首先创建一个函数，计算声压的幅值、相位关系和间隔以及空间中的远域位置。

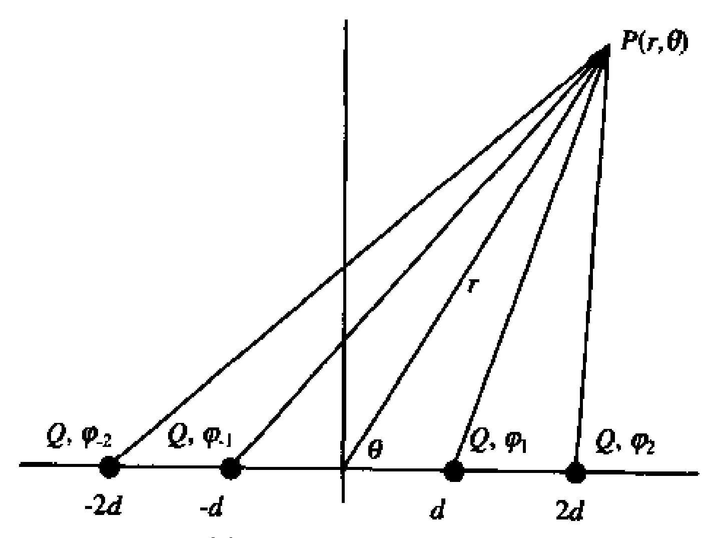


图 6.31 声源的线性排列

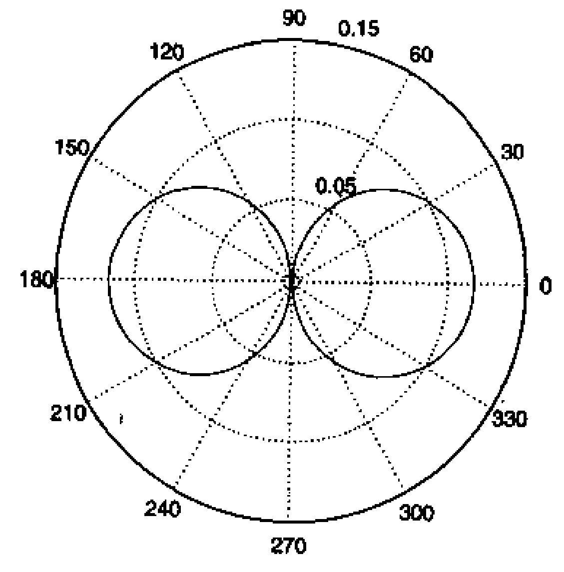


图 6.32 声音偶极子的辐射模式

6.16 时变信号的概率密度函数用于描述周期 T 内信号振幅介于 x 和 $x+dx$ 之间的可能性。换言之，它是用来得到幅值范围内信号时间的百分比。概率密度函数可近似为：

$$P(x) = \lim_{\substack{\Delta x \to 0 \\ T \to \infty}} \left[\sum_i \Delta t_i / T\Delta x \right]$$

表达式中的各项在图 6.33 中标出，图 6.33 画出了正弦波的一个周期。振幅为 A_0 的正弦波的概率密度函数 $P(x)$ 如下：

$$P(x) = \frac{1}{\pi\sqrt{A_o^2 - x^2}} \qquad |x| \leq A_o$$

$$= 0 \qquad |x| > A_o$$

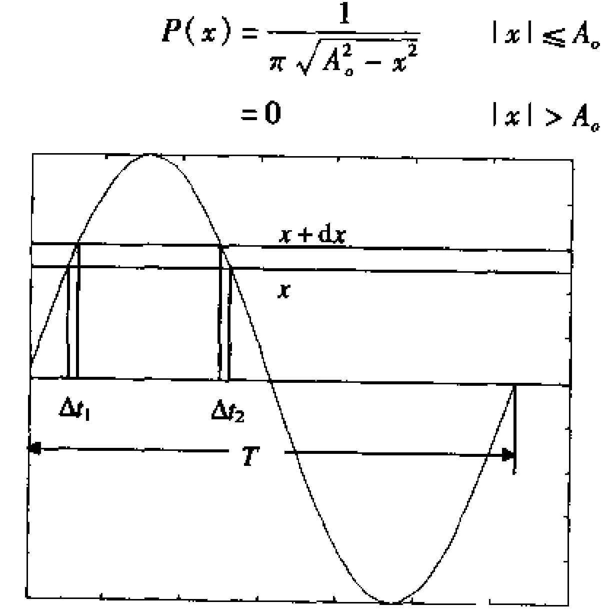


图 6.33 求出正弦波在 x 和 $x+dx$ 之间所经的时间

估算概率密度函数：

$$y = A_o \sin t$$

在 $-\pi \leq t \leq \pi$ 时的结果并将其与实际值相比较。令 $A_o = 2$，振幅区间数等于 20，时间间隔内的数值点数为 400，画出 $P(x)$ 的估计值和实际值。结果如图 6.34 所示，其中的图例已用手工移掉。

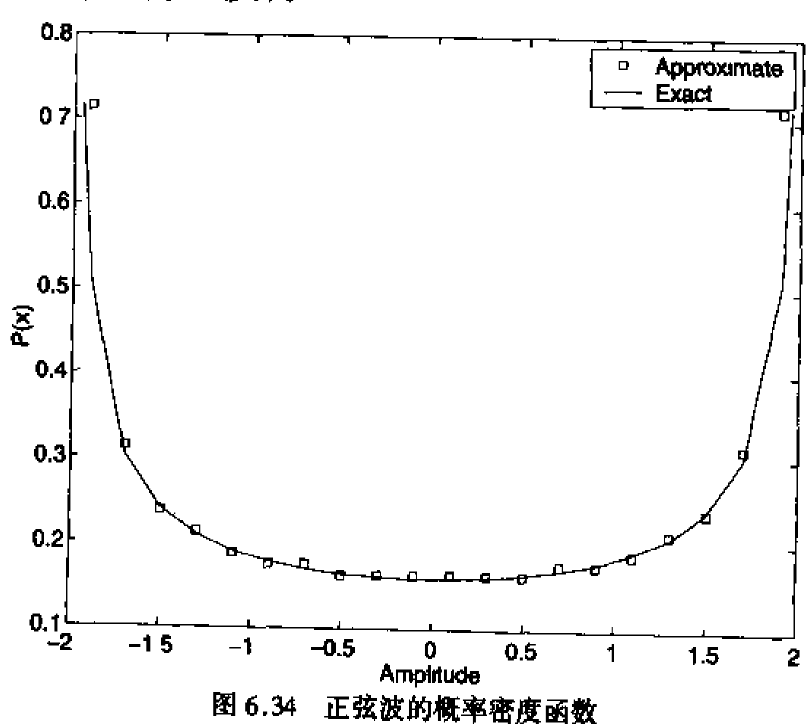


图 6.34 正弦波的概率密度函数

6.17　如图 6.35 所示矩形，由图中看出：

$$r_1=\sqrt{d^2+(W/2)^2}\qquad \alpha=\arctan(W/2d)$$

$$r_2=\sqrt{(d+L)^2+(W/2)^2}\qquad \beta=\arctan(W/2(d+L))$$

如果给出 L、W 和 d 的值，编写一段程序计算重复画在图 6.36 中不相交矩形的最大个数。其中 $L=1$，$W=2$，$d=2$，可生成矩形的最大数目用 floor(π/α)确定。程序未使用 for 循环。

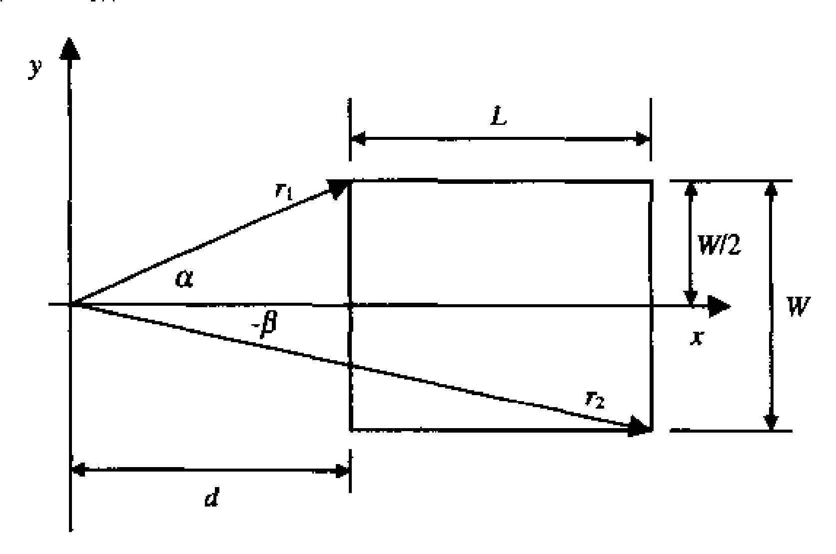


图 6.35　练习 6.17 中的矩形描述

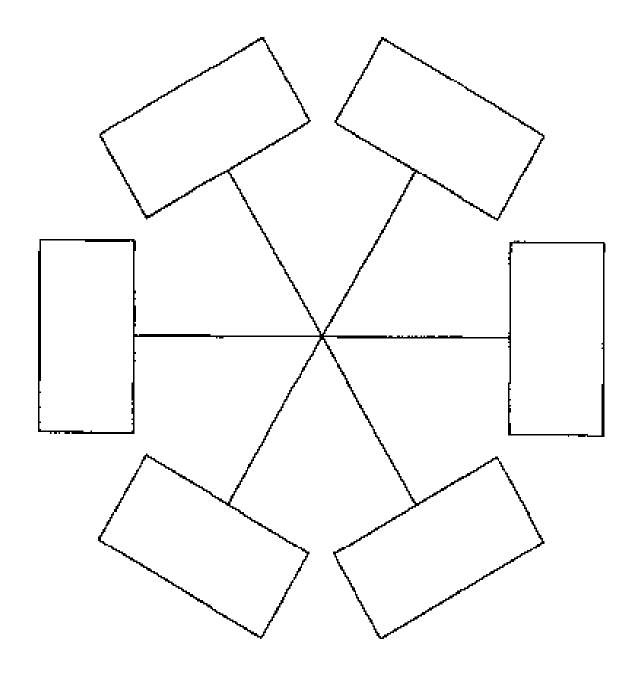


图 6.36　重复绘制的不相交矩形

第7章 三维图形

本章介绍了大部分三维画图功能的实现。

7.1 三维线

3D的plot函数为：

```
plot3(u1,v1,w1,c1,u2,v2,w2,c2,...)
```

其中 u_j,v_j,w_j 分别是某点或一系列点(x,y,z)的坐标。它们可以是3个一组的数、等长向量、等阶次矩阵，或者是在计算中可生成上述三种数值之一的表达式。c_j 是字符串，其中一个字符定义颜色，一个字符定义点的属性，最多两个字符定义线型。见第6.2节关于plot函数的介绍。

假设要画一系列未连接的线，其端点为(x_{1n},y_{1n},z_{1n})和(x_{2n},y_{2n},z_{2n})。为此，创建6个向量：

$$x_j=[x_{j1}\,x_{j2}\quad x_{jn}]$$
$$y_j=[y_{j1}\,y_{j2}\quad y_{jn}]\qquad j=1,2$$
$$z_j=[z_{j1}\,z_{j2}\quad z_{jn}]$$

则plot3函数的指令为：

```
plot3([x1;x2],[y1;y2],[z1;z2])
```

其中[x1;x2],[y1;y2],[z1;z2]均为$(2\times n)$阶矩阵。使用plot函数画出一个三维相似形。

除了用text3代替text函数，用zlabel标注z轴外，其他所有在第6.2节讨论过的二维画图的注释过程均适用于三维曲线及曲面的生成函数。

例7.1 线框盒

如图7.1所示，盒子的尺寸为$L_x\times L_y\times L_z$。创建函数BoxPlot3，画盒子6个面的4条边。盒子的位置和方向由两个对角坐标：$P(x_0,y_0,z_0)$和$P(x_0\pm L_x,y_0\pm L_y,z_0\pm L_z)$确定。

```
function BoxPlot3(x0,y0,z0,Lx,Ly,Lz)
x=[x0   x0      x0      x0      x0+Lx   x0+Lx   x0+Lx   x0+Lx ];
y=[y0   y0      y0+Ly   y0+Ly   y0      y0      y0+Ly   y0+Ly ];
z=[z0   z0+Lz   z0+Lz   z0      z0      z0+Lz   z0+Lz   z0    ];
index = zeros(6,5);
index(1,:)=[1 2 3 4 1];
index(2,:)=[5 6 7 8 5];
index(3,:)=[1 2 6 5 1];
index(4,:)=[4 3 7 8 4];
index(5,:)=[2 6 7 3 2];
index(6,:)=[1 5 8 4 1];
for k = 1:6
 plot3(x(index(k,:)),y(index(k,:)),z(index(k,:)))
 hold on
end
```

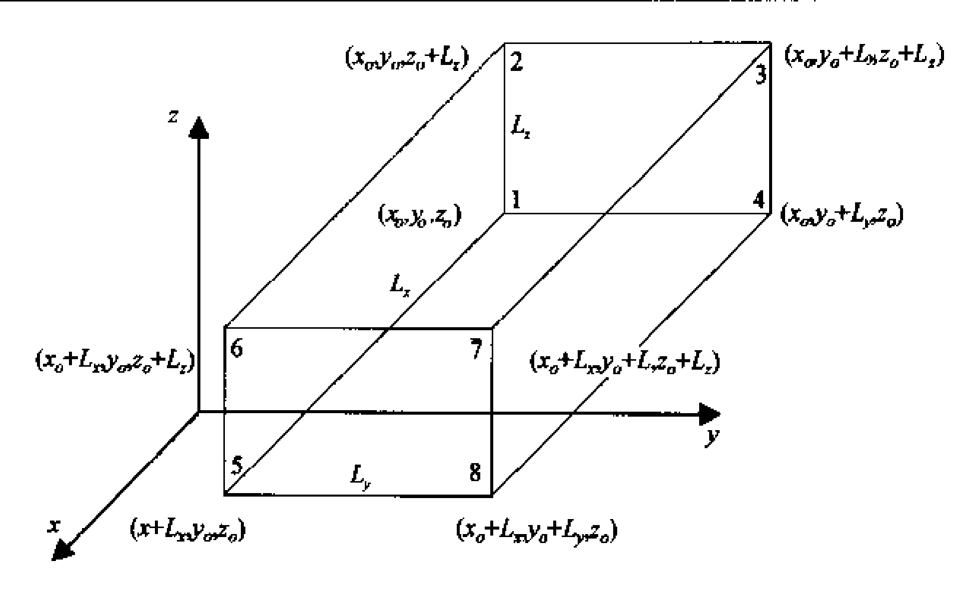


图 7.1 盒的坐标

下述程序创建了三个盒体，并给出其三维尺寸和一个角的坐标。

盒体 #1:

尺寸:3×5×7

坐标:(1,1,1)

盒体 #2:

尺寸:4×5×1

坐标:(3,4,5)

盒体 #3:

尺寸:1×1×1

坐标:(4.5,5.5,6)

创建和显示这些线框盒的程序如下：

```
BoxPlot3(1,1,1,3,5,7)
BoxPlot3(4,6,8,4,5,1)
BoxPlot3(8,11,9,1,1,1)
```

结果如图 7.2 所示。

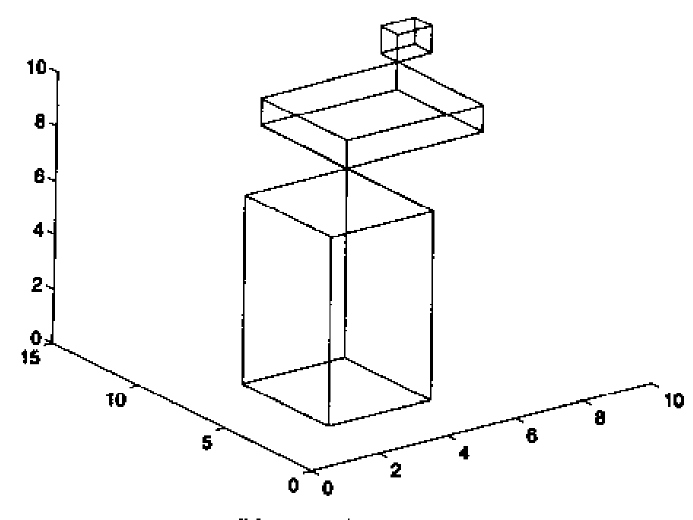


图 7.2 线框盒

7.2 曲面

更强的三维画图函数可生成曲面图、三维等高线图、立体图以及这些基本图形的变形和特殊化处理。基本的曲面绘图函数为：

```
surf
```

和

```
mesh
```

函数 surf 画出的曲面由彩块组成，而 mesh 画出由边界给定的白色曲面。在 surf 函数中，块的颜色由 z 轴方向的值决定，而 mesh 中的线颜色也由 z 轴数值决定。

曲面的定义表达式为：

$$z = f(x, y)$$

其中 x, y 是 xy 平面的坐标，z 是得出的高度，生成三维曲面的语句为：

```
surf(x,y,z)
```

surf 函数用于生成图 7.3，而 mesh 函数用于生成图 7.6。下面将在创建下式所确定的三维曲面过程中进一步介绍上述两个函数和其他几个函数的用法。

$$z(x_1, x_2) = x_1^4 + 3x_1^2 + x_2^2 - 2x_1 - 2x_2 - 2x_1^2 x_2 + 6$$

其范围为 $-3 < x_1 < 3$, $-3 < x_2 < 13$。程序如下：

```
xx1 = linspace(-3,3,15);                                        %(1×15)
xx2 = linspace(-3,13,17);                                       %(1×17)
[x1,x2] = meshgrid(xx1,xx2);                                    %(17×15)
z = x1.^4 + 3 * x1.^2 - 2 * x1 + 6 - 2 * x2. * x1.^2 + x2.^2 - 2 * x2;  %(17×15)
surf(x1,x2,z)
```

meshgrid 函数创建了两个(17×15)阶矩阵，因此可以估算出 x_1 和 x_2 在任意组合下 z 的值。估算方法是通过计算 $z = z(x_1, x_2)$ 时由隐含的点操作完成的。

程序执行结果如图 7.3 所示。在显示曲面的同时为显示出投影到 $z = 0$ 平面上的几条等高线，用 surfc 代替 surf，得到如图 7.4 所示的结果。surfc 和 surf 的参数相同。

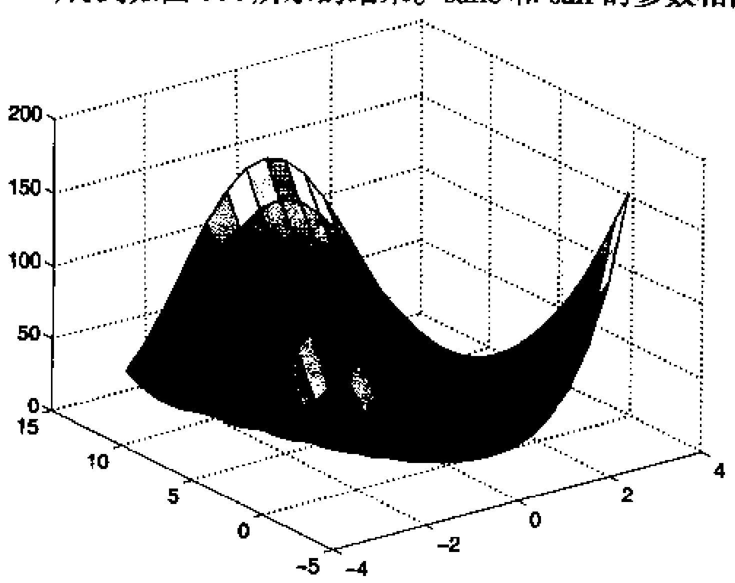


图 7.3　用 surf 函数生成的曲面图

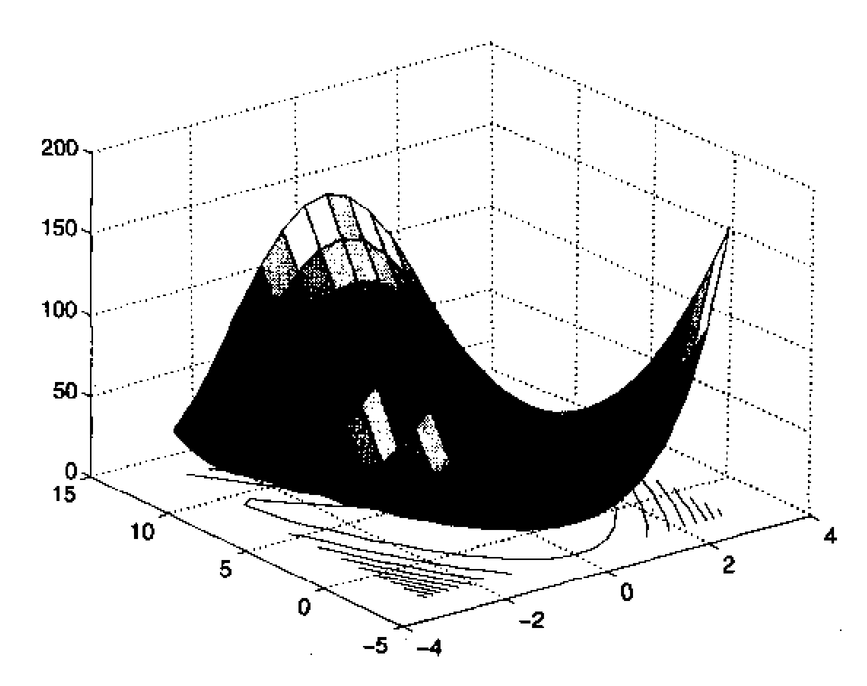


图 7.4 用 surfc 函数生成的曲面及等高线图

从曲面的四个角分别画直线到 $z=0$ 平面，这些角的顶点坐标为：

$$[-3,-3,z(-3,-3)]$$

$$[-3,13,z(-3,13)]$$

$$[3,13,z(3,13)]$$

$$[3,-3,z(3,-3)]$$

程序为：

```
xx1 = linspace( -3,3,15);
xx2 = linspace( -3,13,17);
[x1,x2] = meshgrid(xx1,xx2);
z = x1.^4 + 3 * x1.^2 - 2 * x1 + 6 - 2 * x2. * x1.^2 + x2.^2 - 2 * x2;
surf(x1,x2,z)
x11 = [ -3  -3 3 3];
x22 = [ -3 13 13  -3];
z2 = x11.^4 + 3 * x11.^2 - 2 * x11 + 6 - 2 * x22. * x11 ^2 + x22.^2 - 2 * x22;
hold on
plot3([x11;x11],[x22;x22],[zeros(1,4);z2],'b')
```

程序执行结果如图 7.5 所示。第四条线被图形遮住。

如前所述，用 surf 函数生成图形的颜色与 z 值有关。与之类似，用 mesh 函数绘制图形时，自动产生的线颜色也随 z 值的不同而不同。块或线的颜色可用：

```
colormap(c)
```

语句设置为常数，其中 c 是一个在 0 和 1 之间变化的三元向量。第一个元素对应红色强度，第二个元素对应绿色强度，第三个元素对应蓝色强度。参见帮助文件中 colormap 函数的说明。一些常用颜色的组合为：

$c=[0\ 0\ 0]$→黑

$c=[1\ 1\ 1]$→白

$c=[1\ 0\ 0]$→红

$c=[0\ 1\ 0]$→绿

$c=[0\ 0\ 1]$→蓝

$c=[1\ 1\ 0]$→黄

$c=[1\ 0\ 1]$→品红

$c=[0\ 1\ 1]$→青

$c=[0.5\ 0.5\ 0.5]$→灰

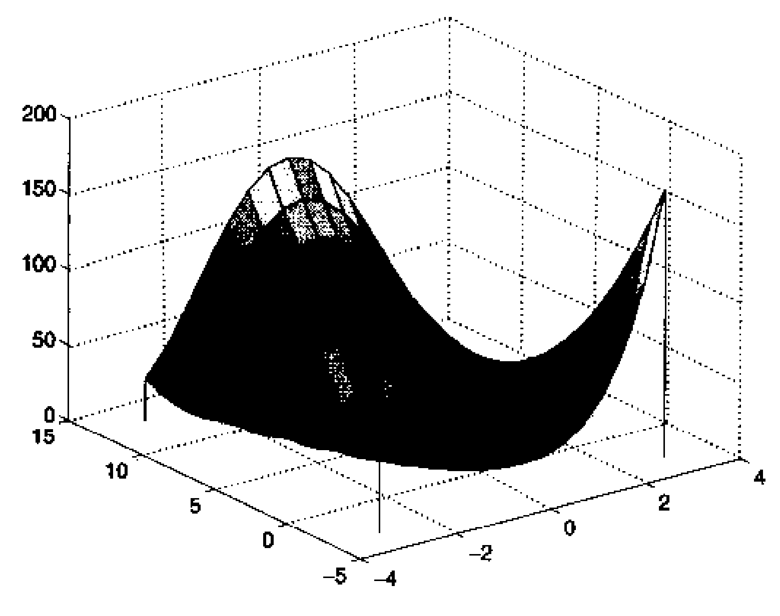


图 7.5 曲面图及四角的垂线

为说明 surf 与 mesh 之间的区别及 colormap 函数的用法,用下述语句代替前述程序的 surf 函数:

```
mesh(x1,x2,z)
colormap([0 0 1])
```

得到图 7.6,且图中所有线均为蓝色。

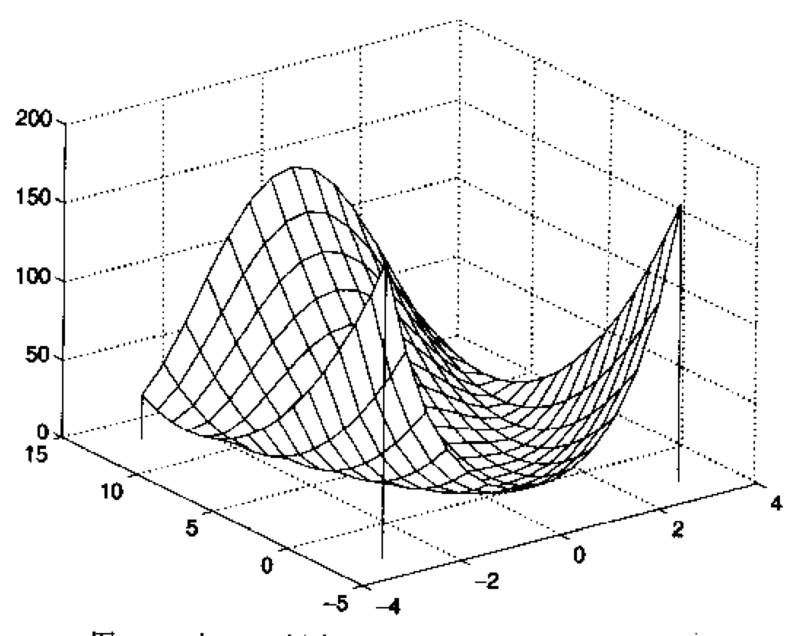


图 7.6 由 mesh 创建的曲面并用 colormap 定义其颜色

函数：

```
grid off
```

可移去图形中的网格线。恢复原来的设置，可用：

```
grid on
```

另外，用：

```
box on
```

语句可在图形四周放置一个盒子。因此，修改图 7.6，移去图形中的网格线并在其四周放置一个盒子。程序如下：

```
xx1 = linspace( -3,3,15);
xx2 = linspace( -3,13,17);
[x1,x2] = meshgrid(xx1,xx2);
z = x1.^4 + 3 * x1.^2 - 2 * x1 + 6 - 2 * x2. * x1.^2 + x2.^2 - 2 * x2;
mesh(x1,x2,z)
colormap([0 0 1])
x11 = [ -3  -3 3 3];
x22 = [ -3 13 13  -3];
z2 = x11.^4 + 3 * x11.^2 - 2 * x11 + 6 - 2 * x22  * x11.^2 + x22.^2 - 2 * x22;
hold on
plot3([x11,x11],[x22,x22],[zeros(1,4);z2],'b')
grid off
box on
```

程序执行结果生成图 7.7。此外，如果用 grid off 语句隐藏网格线，并用 box off 和 axis off 语句分别移去盒子和轴线，则可得到图 7.8。

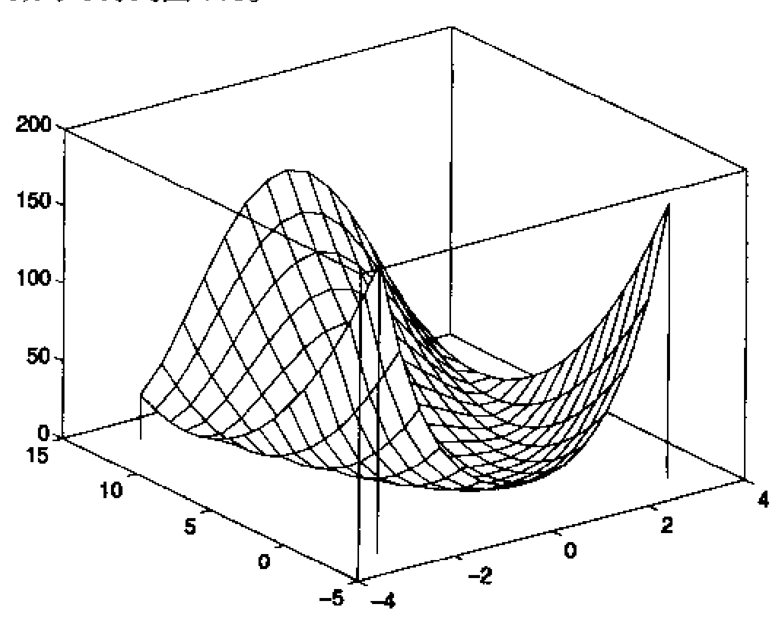


图 7.7 加入 grid off 与 box on 后图 7.6 中的曲面

其他几种加强图 7.3 至图 7.8 中图形视觉效果的方法由图 7.9 至图 7.13 示例给出。图 7.9 由如下程序获得：

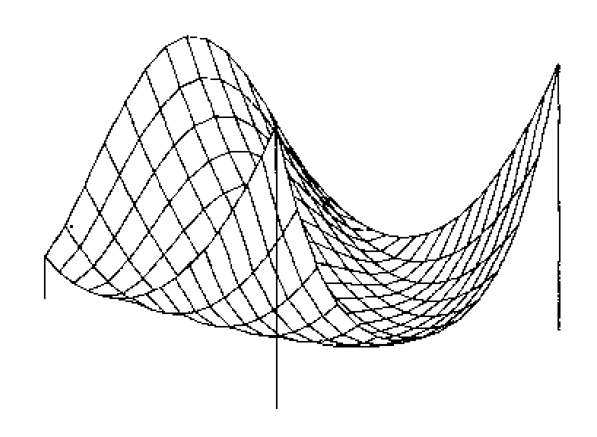


图 7.8 加入 grid off、box off 和 axis off 后图 7.6 中的曲面

```
xx1 = linspace( -3,3,15);
xx2 = linspace( -3,13,17);
[x1,x2] = meshgrid(xx1,xx2);
z = x1.^4 + 3 * x1.^2 - 2 * x1 + 6 - 2 * x2. * x1.^2 + x2.^2 - 2 * x2;
meshz(x1,x2,z)
colormap([0 0 1])
```

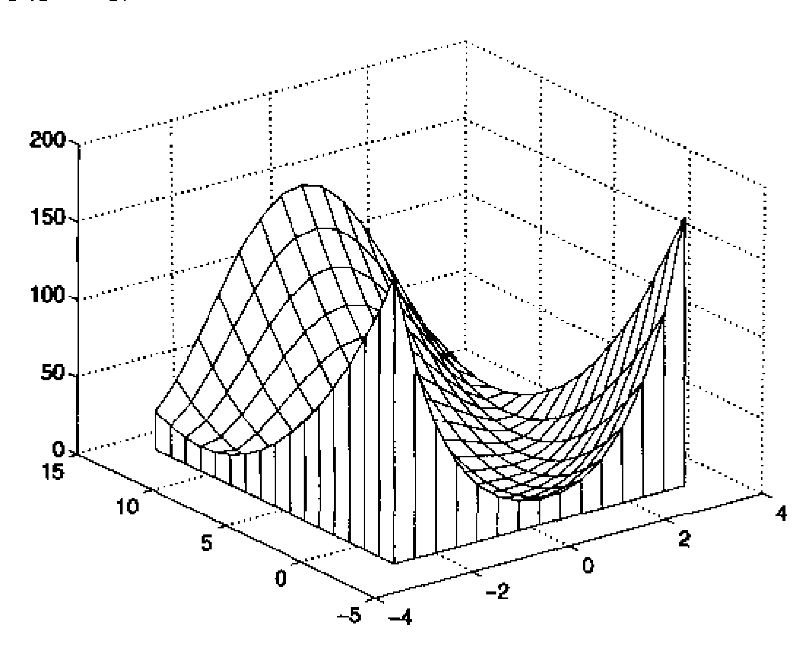


图 7.9 用 meshz 函数创建的曲面

用 waterfall 函数代替上述程序中的 meshz 函数可得到图 7.10。waterfall 函数的参数与 meshz 函数中的参数相同。

曲面也可通过如下语句转换为不同的等高线图：用 contour 得到二维等高线，用 contour3 得到三维等高线，用 contourf 得到填充的等高线。

在 contour 和 contour3 语句之后应用 clabel 语句，可对其图形进行标注。contourf 可用 colorbar 语句标注，colorbar 生成一个彩色棒图，并在图上标出相关数值。下面说明这些函数的用法。

为得到图 7.4 所示的二维等高线，编写如下程序：

```
xx1 = linspace( -3,3,15);
xx2 = linspace( -3,13,17);
[x1,x2] = meshgrid(xx1,xx2);
z = x1.^4 + 3 * x1.^2 - 2 * x1 + 6 - 2 * x2. * x1.^2 + x2.^2 - 2 * x2;
```

```
h = contour(x1,x2,z);
clabel(h)
```

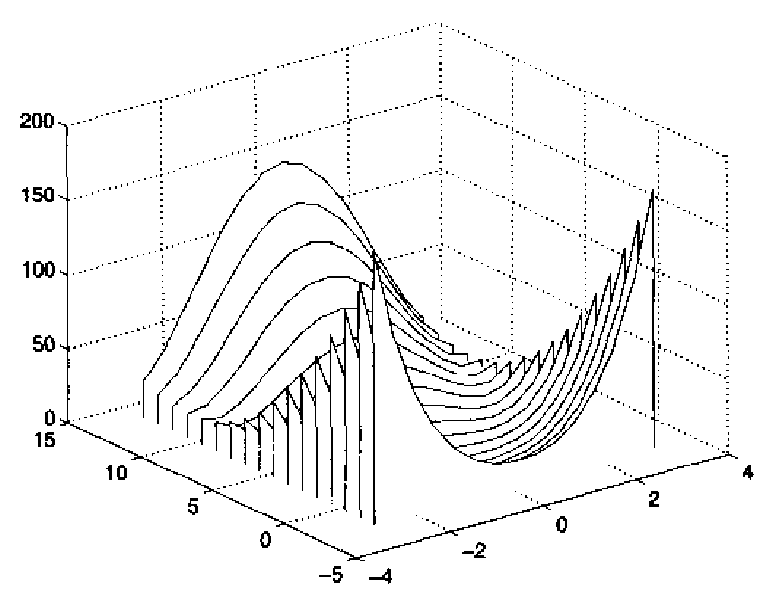


图 7.10 用 waterfall 函数生成的曲面

程序执行结果生成图 7.11。当使用 clabel 函数时，应使用 h = contour(…)语句，否则等高线没有数值显示。如果用 contour3 代替 contour，则得到图 7.12。contour 与 contour3 的参数相同。

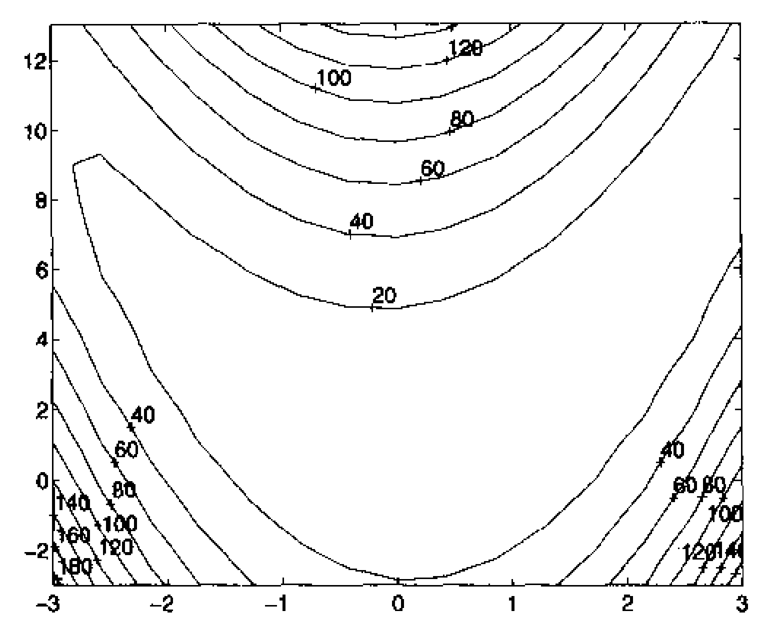


图 7.11 用 contour 和 clabel 标注的等高线

为得到用彩色填充的等高线图，使用 contourf 和 colorbar 函数，程序如下：

```
xx1 = linspace( -3,3,15);
xx2 = linspace( -3,13,17);
[x1,x2] = meshgrid(xx1,xx2);
z = x1.^4 + 3 * x1.^2 - 2 * x1 + 6 - 2 * x2. * x1.^2 + x2.^2 - 2 * x2;
contourf(x1,x2,z);
colorbar('vert')
```

其中 vert 使 colorbar 以垂直方式显示彩色棒图。程序执行结果如图 7.13 所示。

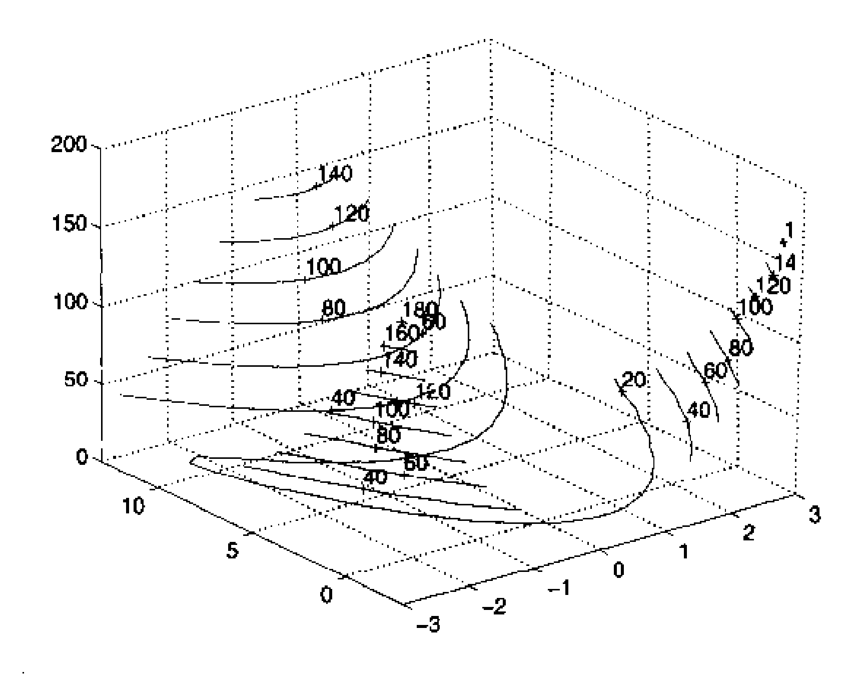


图7.12　用contour3和clabel标注三维等高线图

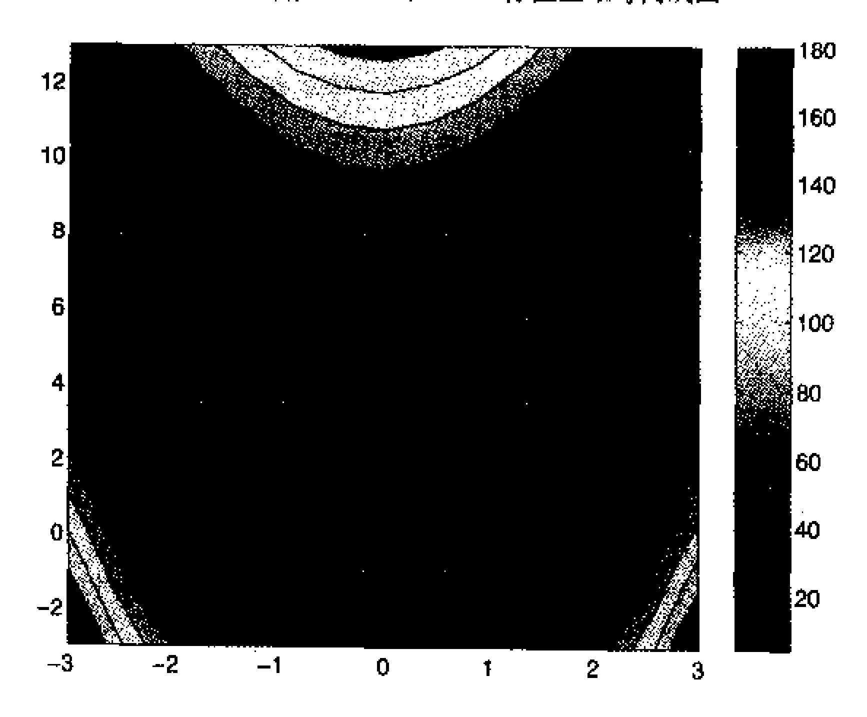


图7.13　函数contourf和colorbar的应用

可以用函数cylinder将一条二维曲线进行旋转，生成一个立体图形。为此，考虑如下曲线：

$$y = 1.1 + \sin(z) \quad 0 \leqslant z \leqslant 2\pi$$

将其沿 z 轴旋转360°。程序如下：

```
[x,y,z] = cylinder(1.1 + sin(0:0.25:2 * pi),16);
surf(x,y,z)
axis off
```

程序执行结果如图7.14所示。

图7.14中，视角为默认值。然而，有时基于下述几点原因，希望改变其默认视角：(1)图形未显示出期望的曲面特征；(2)希望用subplot函数显示曲面的几种不同视图；(3)决定最后结

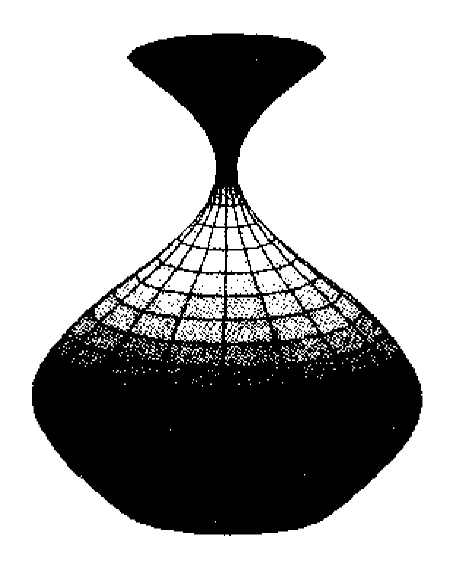


图 7.14 函数 cylinder 的应用

果之前希望对曲面进行多视角观察。MATLAB 提供了直接确定视图方向的方法，就是使用 rotate3d 和 view。图 7.15 中，视角由以下方式决定。在 MATLAB 命令窗口中，键入 rotate3d 语句或在图形窗口中点击旋转图标。操作通过鼠标左键实现。在图形窗口中，按下鼠标左键，将轴拖动到另一位置以增强图形的视觉效果，松开鼠标左键生成三维图形。如果对结果不满意，可重复上述过程直到找到合适的方向。找到合适的方向后，返回 MATLAB 窗口并输入如下语句：

```
[az,elev] = view
```

显示两个数字：第一个是 az，第二个是 elev(可使用其他任何变量名)。因此，可在程序文件中曲面创建函数之后插入：

```
view(n,m)
```

其中 n 是 az 的数值，m 是 $elev$ 的数值。插入语句后发现 $az = -35.5, elev = -34$。修改后的程序为：

```
[x,y,z] = cylinder(1 + sin(0:0.25:2 * pi),16);
surf(x,y,z)
view( - 35.5, - 34)
axis off
```

结果如图 7.15 所示。

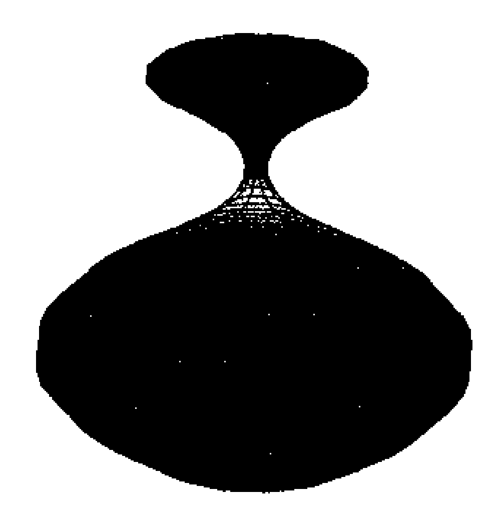


图 7.15 使用 rotate3d 和 view 后的图 7.14

也可在同一图形中创建多个曲面。考虑下述曲面：

$$z(r,\theta)=r^3\cos(3\theta)$$

其中$0\leqslant r\leqslant 1, 0\leqslant\theta\leqslant 2\pi$。假设曲面与两个半径为1的平行圆盘相交于$z=\pm 0.5$，则创建曲面的程序如下：

```
nr = 12; nth = 50;
r = linspace(0,1,nr);
theta = linspace(0,2 * pi,nth);
[R,T] = meshgrid(r,theta)
x = cos(theta') * r;
y = sin(theta') * r;
surf(x,y,R.^3. * cos(3 * T))
hold on
z0 = repmat(0.5,size(x));
surf(x,y,z0)
surf(x,y, - z0)
view( - 42.5,20)
```

结果如图7.16所示。

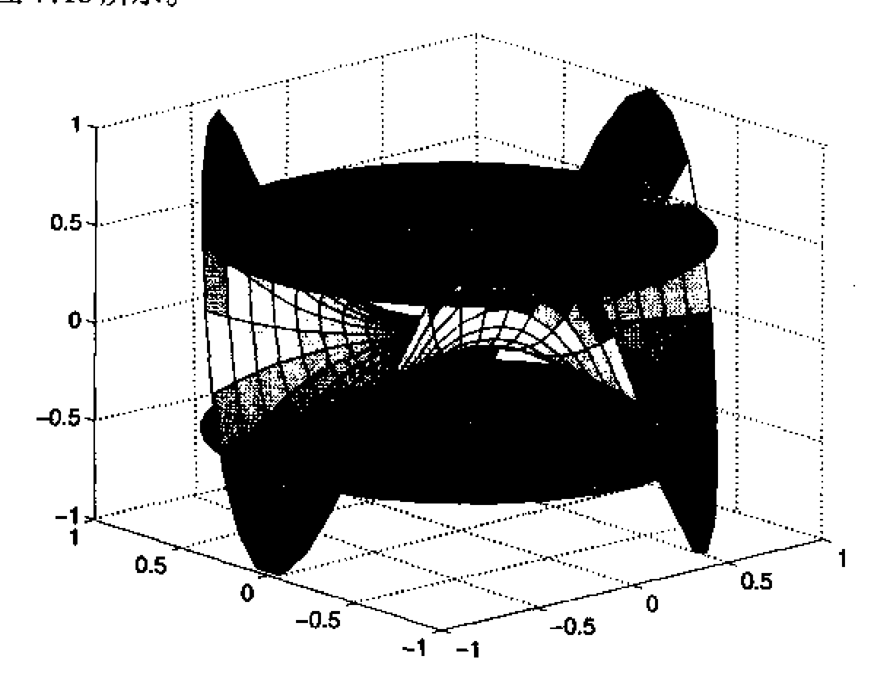


图7.16 一个曲面与两个圆盘相交

例7.2 平面的生成

当空间内的3个点：$\mathbf{P}_0(x_0,y_0,z_0)$，$\mathbf{P}_1(x_1,y_1,z_1)$，$\mathbf{P}_2(x_2,y_2,z_2)$确定时，则包含此三点的平面参数方程为：

$$\mathbf{P}=\mathbf{P}_0+s\mathbf{v}+t\mathbf{w}$$

其中，

$$\mathbf{P}=x\mathbf{i}+y\mathbf{j}+z\mathbf{k}$$

$$\mathbf{P}_0=x_0\mathbf{i}+y_0\mathbf{j}+z_0\mathbf{k}$$

$$\mathbf{v}=v_1\mathbf{i}+v_2\mathbf{j}+v_3\mathbf{k}=(x_1-x_0)\mathbf{i}+(y_1-y_0)\mathbf{j}+(z_1-z_0)\mathbf{k}$$

$$\mathbf{w}=w_1\mathbf{i}+w_2\mathbf{j}+w_3\mathbf{k}=(x_2-x_0)\mathbf{i}+(y_2-y_0)\mathbf{j}+(z_2-z_0)\mathbf{k}$$

$0\leqslant s\leqslant 1, 0\leqslant t\leqslant 1$。因此，

$$x = x_0 + sv_1 + tw_1 = x_0 + s(x_1 - x_0) + t(x_2 - x_0)$$
$$y = y_0 + sv_2 + tw_2 = y_0 + s(y_1 - y_0) + t(y_2 - y_0)$$
$$z = z_0 + sv_3 + tw_3 = z_0 + s(z_1 - z_0) + t(z_2 - z_0)$$

假设使用 5×5 的网格能充分显示该平面,则可创建 PlanarSurface 函数以生成并显示此平面。

```
function PlanarSurface(P0,P1,P2)
v = P1 - P0;
W = P2 - P0;
S = 0:0.2:1;
[s,t] = meshgrid(S,S)
xx = P0(1) + s * v(1) + t * w(1);
yy = P0(2) + s * v(2) + t * w(2);
zz = P0(3) + s * v(3) + t * w(3);
surf(xx,yy,zz)
hold on
```

其中 P_0, P_1, P_2 是表示平面上 3 点坐标的 3 元素向量。因此,如果在 MATLAB 命令窗口输入如下语句:

```
PlanarSurface([0 0 0 ],[2 6 3],[7 1 5])
```

则可得到图 7.17。

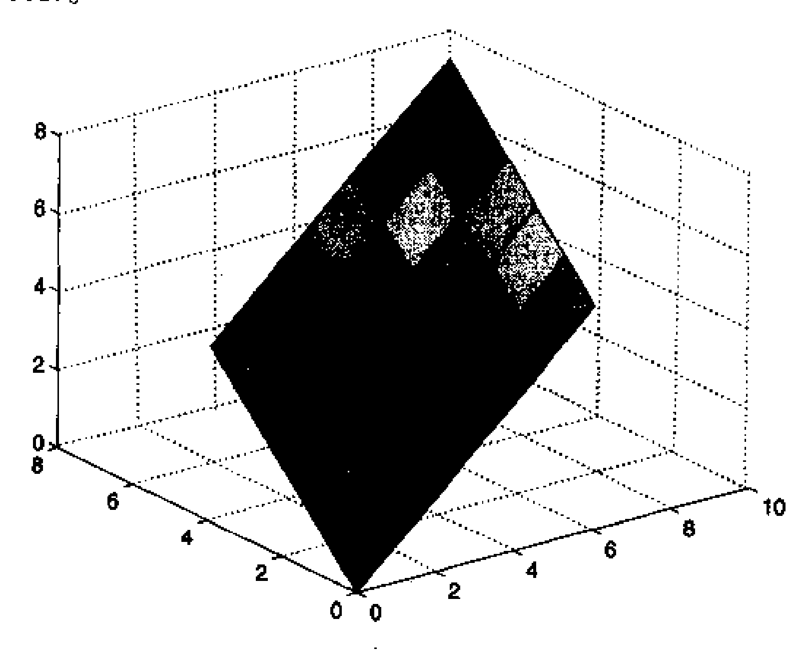


图 7.17 平面的生成

为将平面投影到三个相互垂直的坐标参考面上,应采用适当的点乘运算。因此,为将平面投影到 xy 平面上,则:

$$\mathbf{P}\cdot(\mathbf{i}+\mathbf{j}+0\mathbf{k})$$

该式等价于将平面上所有点的 z 坐标设为最小值。与之类似,如将平面投影到 yz 平面上,则:

$$\mathbf{P}\cdot(0\mathbf{i}+\mathbf{j}+\mathbf{k})$$

该式等价于将平面上所有点的 x 坐标设为可显示的 x 坐标的最大值。如将平面投影到 xz 平面上,则:

$$\mathbf{P}\cdot(\mathbf{i}+0\mathbf{j}+\mathbf{k})$$

该式等价于将平面上所有点的 y 坐标设为可显示的 y 坐标的最大值。为此，修改 *Surface-Planar* 以使其包含投影面的选择。

```
function PlanarSurface(P0,P1,P2,projection)
v = P1 - P0;
w = P2 - P0;
S = 0:0.2:1;L = length(S);
[s,t] = meshgrid(S,S);
xx = P0(1) + s * v(1) + t * w(1);
yy = P0(2) + s * v(2) + t * w(2);
zz = P0(3) + s * v(3) + t * w(3);
surf(xx,yy,zz)
hold on
if nargin > 3
 a = axis;
 c(1:L,1:L,1:3) = zeros(L,L,3);
 c(:,:,1) = 1;
 c(:,:,2) = 1;
 c(:,:,3) = 0;
 surf(xx,yy,a(5) * ones(L,L),c)
 surf(xx,a(4) * ones(L,L),zz,c)
 surf(a(2) * ones(L,L),yy,zz,z)
end
```

其中 *projection* 可以是任何数，数组 c 的定义使投影面显示为黄色。数组 c 的前两个索引必须具有相同阶次，如 *xx*，*yy*，*zz*。最后的索引准确地代表了 3 个元素，其中每个元素的取值范围都从 0 到 1 。最后三个元素定义了前两个索引的每种组合下每个色块的颜色。在 MATLAB 命令窗口键入：

```
PlanarSurface([0 0 0],[2 6 3],[7 1 5],1)
```

结果得到图 7.18。图中似乎给人一种错觉，即投影面并不在设计面上，但用 rotate3d 可证实它们确实在相应的坐标平面上。

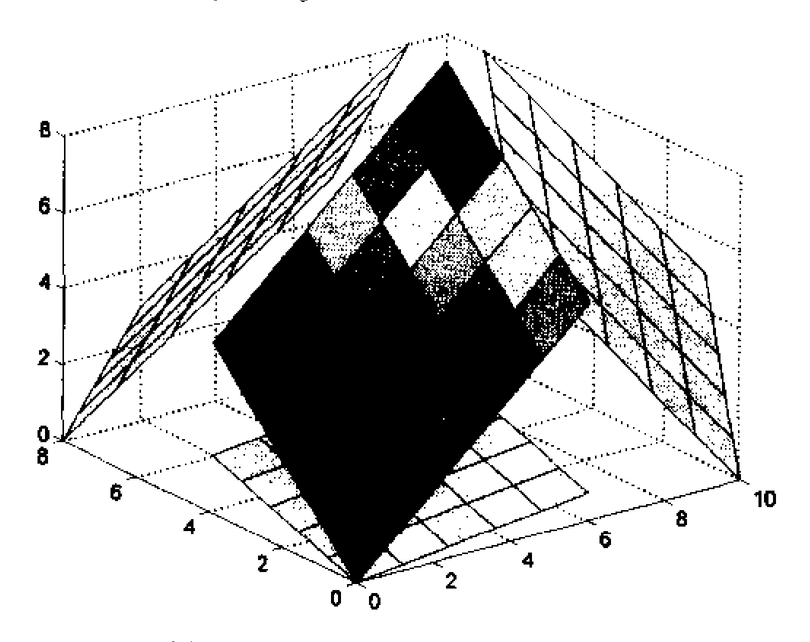


图 7.18 将一个平面投影到参考坐标面上

例 7.3 盒子的生成

用例 7.2 的结果生成长方体(盒子),其尺寸为 $L_x \times L_y \times L_z$。参见图 7.1,由图中可看出盒子由 6 个面组成,且每个面由下述 3 组点定义。

垂直于 yz 平面的面:

$\mathbf{P}_0(x_0, y_0, z_0), \mathbf{P}_1(x_0, y_0, z_0 + L_z), \mathbf{P}_2(x_0, y_0 + L_y, z_0)$

$\mathbf{P}_0(x_0 + L_x, y_0, z_0), \mathbf{P}_1(x_0 + L_x, y_0, z_0 + L_z), \mathbf{P}_2(x_0 + L_x, y_0 + L_y, z_0)$

垂直于 xz 平面的面:

$\mathbf{P}_0(x_0, y_0, z_0), \mathbf{P}_1(x_0, y_0, z_0 + L_z), \mathbf{P}_2(x_0 + L_y, y_0, z_0)$

$\mathbf{P}_0(x_0, y_0 + L_y, z_0), \mathbf{P}_1(x_0, y_0 + L_y, z_0 + L_z), \mathbf{P}_2(x_0 + L_x, y_0 + L_y, z_0)$

垂直于 xy 平面的面:

$\mathbf{P}_0(x_0, y_0, z_0), \mathbf{P}_1(x_0 + L_x, y_0, z_0), \mathbf{P}_2(x_0, y_0 + L_y, z_0)$

$\mathbf{P}_0(x_0, y_0, z_0 + L_z), \mathbf{P}_1(x_0 + L_x, y_0, z_0 + L_z), \mathbf{P}_2(x_0, y_0 + L_y, z_0 + L_z)$

在函数 *BoxSurface* 中应用上述关系:

```
function BoxSurface(P0,L)
PlanarSurface(P0,P0+[0 0 L(3)],P0+[0 L(2) 0])
PlanarSurface(P0+[L(1) 0 0],P0+[L(1) 0 L(3)],P0+[L(1) L(2) 0])
PlanarSurface(P0,P0+[0 0 L(3)],P0+[L(1) 0 0])
PlanarSurface(P0+[0 L(2) 0],P0+[0 L(2) L(3)],P0+[L(1) L(2) 0])
PlanarSurface(P0,P0+[L(1) 0 0],P0+[0 L(2) 0])
PlanarSurface(P0+[0 0 L(3)],P0+[L(1) 0 L(3)],P0+[0 L(2) L(3)])
```

其中 $\mathbf{P}_0 = [x_0, y_0, z_0]$,$L = [L_x, L_y, L_z]$,*PlanarSurface* 在例 7.2 中给出。

下述程序生成三个盒子,盒子的尺寸及其中一个角的位置如下所示:

盒子 #1:

尺寸:3×5×7

位置:(1,1,1)

盒子 #2:

尺寸:4×5×1

位置:(3,4,5)

盒子 #3:

尺寸:1×1×1

位置:(4.5,5.5,6)

创建并显示这些盒子的程序如下:

```
BoxSurface([1,1,1],[3,5,7])
BoxSurface([3,4,5],[4,5,1])
BoxSurface([4,5,5.5,6],[1,1,1])
view(29.5,44)
```

结果如图 7.19 所示。

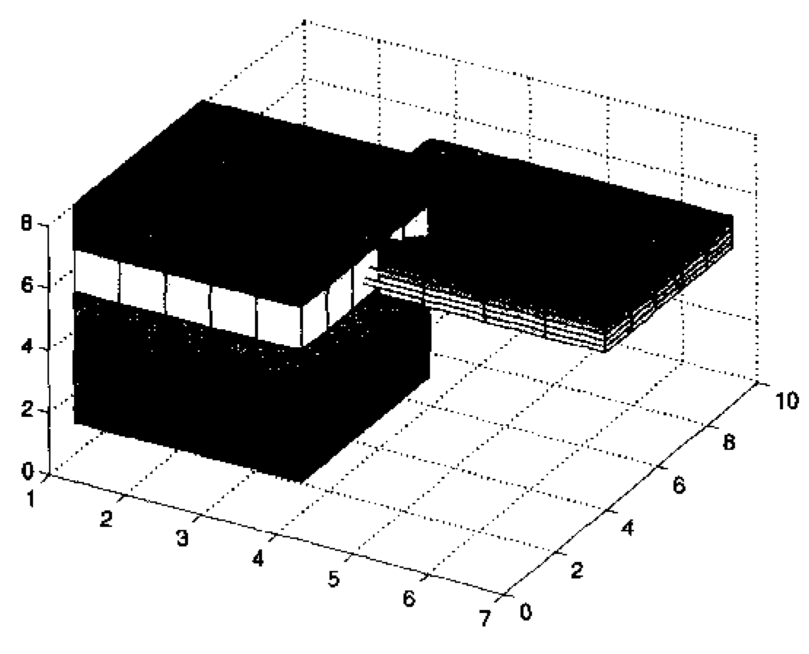


图7.19 三个盒子

例7.4 三维物体的旋转和转换:汽车底盘

从一点 $p(x,y,z)$ 到另一点 $P(X,Y,Z)$ 的旋转和转换为[1]:

$$X = L_x + a_{11}x + a_{12}y + a_{13}z$$

$$Y = L_y + a_{21}x + a_{22}y + a_{23}z$$

$$Z = L_z + a_{21}x + a_{22}y + a_{23}z$$

其中 L_x, L_y, L_z 分别为转换的 (x,y,z) 分量,a_{ij} 为下述矩阵的元素:

$$a = \begin{bmatrix} \cos\Psi\cos\chi & -\cos\Psi\sin\chi & \sin\Psi \\ \cos\phi\sin\chi + \sin\phi\sin\Psi\cos\chi & \cos\phi\cos\chi - \sin\phi\sin\Psi\sin\chi & -\sin\phi\cos\Psi \\ \sin\phi\sin\chi - \cos\phi\sin\Psi\cos\chi & \sin\phi\cos\chi + \cos\phi\sin\Psi\sin\chi & \cos\phi\cos\Psi \end{bmatrix}$$

ϕ, Ψ, χ 是坐标系关于原点的顺序旋转角(欧拉角):ϕ 为 x 轴的旋转角,Ψ 为 y 轴的旋转角,χ 为 z 轴的旋转角。一般而言,(x,y,z) 可以是标量、等长向量或等阶次矩阵。

在应用这些关系之前,先创建如下函数:

```
function [Xrt,Yrt,Zrt] = EulerAngles(psi,chi,phi,Lx,Ly,Lz,x,y,z)
R = [cos(psi) * cos(chi), - cos(psi) * sin(chi), sin(psi);
    cos(phi) * sin(chi) + sin(phi) * sin(psi) * cos(chi),
    cos(phi) * cos(chi) - sin(phi) * sin(psi) * sin(chi), - sin(phi) * cos(psi);
    sin(phi) * sin(chi) - cos(phi) * sin(psi) * cos(chi),
    sin(phi) * cos(chi) + cos(phi) * sin(psi) * sin(chi), cos(phi)(cos(psi)];
Xrt = R(1,1) * x + R(1,2) * y + R(1,3) * z + Lx;
Yrt = R(2,1) * x + R(2,2) * y + R(2,3) * z + Ly;
Zrt = R(3,1) * x + R(3,2) * y + R(3,3) * z + Lz;
```

现在以对圆环面的变换为例,说明这些变换方程的用法,圆环面代表汽车轮胎。它表示为:

---

[1] 参见 W. Gellert, H. Kustner, M. Hellwich 和 H. Kastner, *The VNR Concise Encyclopedia of Mathematics*, Van Nostrand Reinhold, New York, 1975 年,第 534~535 页。

$$z = \pm\sqrt{a^2 - \left(\sqrt{x^2 + y^2} - b\right)^2}$$

其中，

$$x = r\cos\theta$$
$$y = r\sin\theta$$

$b - a \leqslant r \leqslant b + a, 0 \leqslant \theta \leqslant 2\pi$，且 $b > a$。

首先画出圆环面，然后画出绕 x 轴旋转 60°($\phi = 60°$)的圆环，接着将该圆环绕 y 轴旋转 60°($\Psi = 60°$)。最后，在圆环内放一个圆盘，初始平面为 $z = 0$ 平面，然后将两者同时绕 y 轴旋转 90°($\Psi = 90°$)，结果如图 7.20 所示。编写程序之前，首先创建两个附加函数，第一个生成圆环面：

```
function[X,Y,Z] = CarTire(a,b)
r = linspace(b - a,b + a,10)
th = linspace(0,2 * pi,22)
x = r' * cos(th);
y = r' * sin(th);
z = real (sqrt(a^2 - (sqrt(x.^2 + y.^2) - b).^2));
X = [x x];
Y = [y y];
Z = [z -z];
```

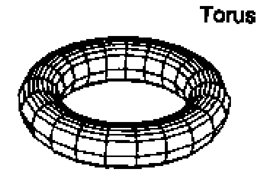
其中 real 用于消除由数值方法产生的虚部。第二个函数生成圆盘：

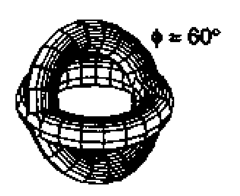
```
function[XD,YD,ZD] = CarDisc(a,b)
discr = linspace(0,b - a,7);
th = linspace(0,2 * pi,16);
XD = discr' * cos(th);
YD = discr' * sin(th);
ZD = zeros(7,16)
```

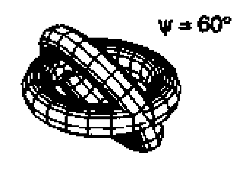
对于 $a = 0.2, b = 0.8$，程序为：

```
a = 0.2;b = 0.8;
[X Y Z] = CarTire(a,b);
[XD YD ZD] = CarDisc(a,b);
Lx = 0;Ly = 0;Lz = 0;
for k = 1:4
 subplot(2,2,k)
 switch k
 case 1
  mesh(X,Y,Z)
  v = axis;
  axis([v(1) v(2) v(3) v(4) -1 1])
  text(0.5,-0.5,1,'Torus')
 case 2
  psi = 0;chi = 0;phi = pi/3;
  [Xr Yr Zr] = EulerAngles(psi,chi,phi,Lx,Ly,Lz,X,Y,Z);
  mesh(X,Y,Z)
  hold on
  mesh(Xr,Yr,Zr)
  text(0.5,-0.5,1,'\phi = 60 \circ')
 case 3
  psi = pi/3;chi = 0;phi = 0;
```

```
    [Xr Yr Zr] = EulerAngles(psi,chi,phi,Lx,Ly,Lz,X,Y,Z);
    mesh(X,Y,Z)
    hold on
    mesh(Xr,Yr,Zr)
    text(0.5,-0.5,1,'\phi=60\circ')
  case 4
    psi=pi/2;chi=0;phi=0;
    [Xr Yr Zr] = EulerAngles(psi,chi,phi,Lx,Ly,Lz,X,Y,Z);
    [Xd Yd Zd] = EulerAngles(psi,chi,phi,Lx,Ly,Lz,XD,YD,ZD);
    mesh(Xd,Yd,Zd)
    hold on
    mesh(Xr,Yr,Zr)
    text(0.5,-0.5,0.7,'\psi=90\circ')
    text(0.5,-0.5,1,'Torus + disc')
  end
  colormap([0 0 1])
  axis equal
  axis off
  grid off
end
```







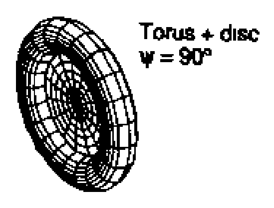


图 7.20 圆环及圆环与内部圆盘的旋转

如图 7.21 所示，利用圆环面及其内置圆盘生成汽车的底盘和轮胎。为此，应生成一个圆柱体，描述如下：

$$x = r\cos\theta$$
$$y = r\sin\theta$$
$$z = z$$

假设圆柱体的直径为 $0.3(b-a)$，其长度依据不同的部位可发生改变。为提高程序的可读性，创建一个附加函数以生成一个长度为 $2L_2$ 的轴：

```
function[XC,YC,ZC] = CarShaft(a,b,L2)
th = linspace(0,2*pi,16);
[XC,ZC] = meshgrid(0.3*(b-a)*cos(th),[-L2 L2]);
YC = meshgrid(0.3*(b-a)*sin(th),[-L2 L2]);
```

程序如下：

```
a=0.2;b=0.8;
[X Y Z]=CarTire(a,b);
[XD YD ZD]=CarDisc(a,b);
psi=[pi/2 pi/2 pi/2 pi/2];chi=[0 0 0 0];phi=[0 0 0 0];
Lx=[3 -3 3 -3];Ly=[0 0 8 8];Lz=[0 0 0 0];
for k=1:4
 [Xr Yr Zr]=EulerAngles(psi(k),chi(k),phi(k),Lx(k),Ly(k),Lz(k),X,Y,Z);
 [Xd Yd Zd]=EulerAngles(psi(k),chi(k),phi(k),Lx(k),Ly(k),Lz(k),XD,YD,ZD);
 mesh(Xd,Yd,Zd)
 hold on
 mesh(Xr,Yr,Zr)
end
psi=[pi/2 pi/2 0 0 atan(0.5) -atan(0.5)];
chi=[0 0 0 0 0 0];
phi=[0 0 pi/2 pi/2 pi/2 pi/2]
Lx=[0 0 2 -2 0 0];Ly=[0 8 4 4 4 4];Lz=[0 0 0 0 0 0];
for k=1:6
 switch k
 case{1,2}
  [XC,YC,ZC]=CarShaft(a,b,[-3 3]);
 case{3,4}
  [XC,YC,ZC]=CarShaft(a,b,[-4 4]);
 case{5,6}
  [XC,YC,ZC]=CarShaft(a,b,[-4.4721 4.4721]);
 end
 [Xc Yc Zc]=EulerAngles(psi(k),chi(k),phi(k),Lx(k),Ly(k),Lz(k),XC,YC,ZC);
 mesh(Xc,Yc,Zc)
end
colormap([0 0 1])
axis equal
axis off
```

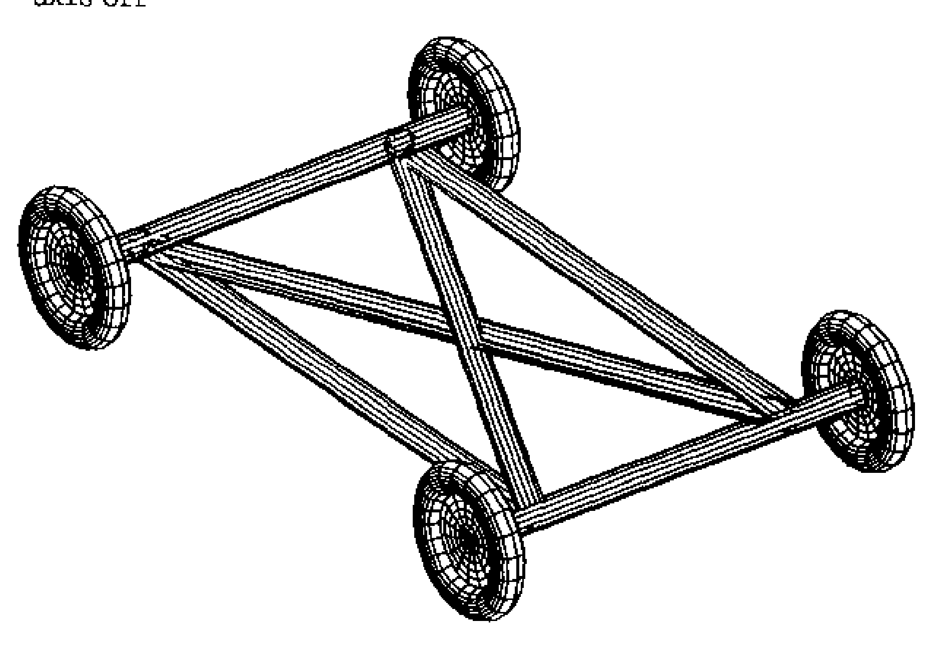


图7.21 汽车底盘

练习

7.1 使用 axis equal 语句画出下述三维曲线[①]。

球面螺旋线($c=5.0, 0\leqslant t\leqslant 10\pi$):

$$x=\sin(t/2c)\cos(t)$$
$$y=\sin(t/2c)\sin(t)$$
$$z=\cos(t/2c)$$

柱面正弦波($a=10.0, b=1.0, c=0.3, 0\leqslant t\leqslant 2\pi$):

$$x=b\cos(t)$$
$$y=b\sin(t)$$
$$z=c\cos(at)$$

球面正弦波($a=10.0, b=1.0, c=0.3, 0\leqslant t\leqslant 2\pi$):

$$x=\cos(t)\sqrt{b^2-c^2\cos^2(at)}$$
$$y=\sin(t)\sqrt{b^2-c^2\cos^2(at)}$$
$$z=c\cos(at)$$

环面螺旋线($a=0.2, b=0.8, c=20.0, 0\leqslant t\leqslant 2\pi$):

$$x=[b+a\sin(ct)]\cos(t)$$
$$y=[b+a\sin(ct)]\sin(t)$$
$$z=a\cos(ct)$$

7.2 画出下列实体的曲面图形[②]。对(a)到(f)中的坐标,适当时可采用坐标变换 $x=r\cos(\theta)$, $y=r\sin(\theta)$或 $x=a\cos(\theta)$, $y=b\sin(\theta)$得到的向量形式,令 r(或 a 与/或 b)为10,步长 θ 为 22。练习(g)要求使用 meshgrid 函数,所有图形要求使用 axis equal 语句。

球体($r=1, 0\leqslant\theta\leqslant 2\pi$):

$$z=\sqrt{1-x^2-y^2}$$

椭圆体($a=1.0, b=1.5, c=2.0$):

$$z=c\sqrt{1-x^2/a^2-y^2/b^2}$$

扁球体(椭圆体 $a=b>c$; $a=b=1.0, c=0.5$)

扁长球体(椭圆体 $a=b<c$; $a=b=1.0, c=1.2$)

圆锥体($-2\leqslant x\leqslant 2, -2\leqslant y\leqslant 2$)[必须使用两次 surf 函数。]:

$$z=\pm\sqrt{x^2+y^2}$$

其中 $x=r\cos\theta$, $y=r\sin\theta$,其取值范围为 $0\leqslant r\leqslant 2, 0\leqslant\theta\leqslant 2\pi$。

---

① 参见 D. von Seggern, *CRC Standard Curves and Surfaces*, CRC Press, Inc., Boca Raton, FL, 1993 年。

② 参见 D. von Seggern,出处同上。

丰饶角($a=0.3,b=0.5,0\leqslant u\leqslant 2\pi,-3\leqslant v\leqslant 3$)[用 rotate3d 分析曲面。]:

$$x=e^{bv}\cos(v)+e^{av}\cos(u)\cos(v)$$
$$y=e^{bv}\sin(v)+e^{av}\cos(u)\sin(v)$$
$$z=e^{av}\sin(u)$$

7.3 画出下列曲面[1],并用 rotate3d 函数进行分析。

(a) ($a=b=1,c=0.5,-3\leqslant x\leqslant 3,-3\leqslant y\leqslant 3$):

$$z=c((x/a)^4\pm(y/b)^4)$$

(b) ($a=3,c=0.25,-1\leqslant x\leqslant 1,-1\leqslant y\leqslant 1$):

$$z=c\sin\left(2\pi a\sqrt{x^2+y^2}\right)$$

(c) ($a=3,c=0.25,-1\leqslant x\leqslant 1,-1\leqslant y\leqslant 1$):

$$z=c\sin(2\pi axy)$$

(d) ($c=0.2,-1\leqslant x\leqslant 1,-1\leqslant y\leqslant 1;x\neq 0,y\neq 0$):

$$z=c\ln(|xy|)$$

(e) 悬索曲面($1\leqslant u\leqslant 5,0\leqslant v\leqslant 2\pi$):

$$x=u\cos(v)$$
$$y=u\sin(v)$$
$$z=\operatorname{arcosh}(u)$$

(f) 正螺旋面($c=1/2\pi,-0.5\leqslant u\leqslant 0.5,-2\pi\leqslant v\leqslant 2\pi$):

$$x=u\cos(v)$$
$$y=u\sin(v)$$
$$z=cv$$

7.4 一个边缘固定的实心圆盘,$r=b$,其模型外形为:[2]

$$w_{mn}(r,\theta)=[C_{mn}J_m(\Omega_{mn}r/b)+I_m(\Omega_{mn}r/b)]\cos(m\theta)$$

其中 $m=0,1,2,\cdots,J_m(x)$为一类 m 阶贝塞尔函数,$I_m(x)$为一类 m 阶贝塞尔函数的修正函数:

$$C_{mn}=-\frac{I_m(\Omega_{mn})}{J_m(\Omega_{mn})}$$

Ω_{mn}为下式的解:

$$J_m(\Omega_{mn})I_{m+1}(\Omega_{mn})+I_m(\Omega_{mn})J_{m+1}(\Omega_{mn})=0$$

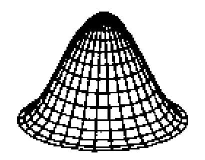
且在练习 5.2(f)中已求出。

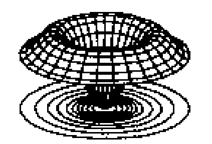
利用练习 5.2(f)中,当 $m=0,1,2$ 时所得到的三个最低自然频率系数,使用 surfc 和 subplot 函数在同一图形中画出相应的 9 种模型外形,并在每个图形的上方标注 m,n 值及相应的频率系数。不要画出 x 轴和y 轴。图形中的第一行为 $m=0$ 时的情形,依此类推。使用两次 max 函数(因为位移域是矩阵)对每个模型的外形进行标准化处理,使其最大幅值为 1。建议半径分数(r/b)为 15,角分数为 30。结果如图 7.22

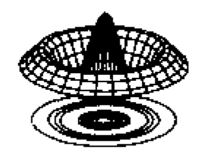
① 参见 D. von Seggern,出处同上。

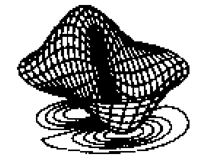
② 参见 E.B Magrab, *Vibration of Elastic Structural Members*, Sijthoff & Noordhoff, The Netherlands, 1979 年,第 252 页。

所示。运用 meshc 函数提高复制图形的清晰度。



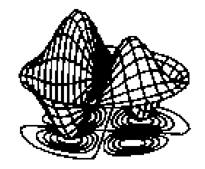


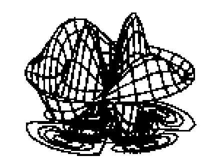












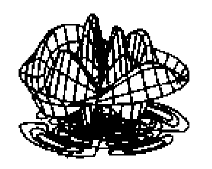


图 7.22　一个圆盘的各种形状

7.5　一块厚平板，x 方向的厚度为 $2L$，y，z 方向均很长。$t=0$ 时它为恒温 T_i。如果突然将厚平板置于温度为 T_∞ 的对流环境中，则其温度分布随时间与位置变化的函数关系为：[①]

$$\frac{\theta}{\theta_i}=2\sum_{n=1}^{\infty}\frac{\sin\delta_n\cos(\delta_n\eta)}{\delta_n+\sin\delta_n\cos(\delta_n)}\exp(-\delta_n^2\tau)$$

其中 $\theta=\theta(\eta,\tau)=T-T_\infty$，$T=T(\eta,\tau)$ 为厚平板温度，$\theta_i=T_i-T_\infty$，$\eta=x/L$，$\tau=a^2t/L^2$ 为无量纲时间(有时称为傅里叶系数)，α 为热扩散系数，δ_n 为下式的解：

$$\cot\delta_n=\frac{\delta_n}{B_i}$$

其中 $B_i=\bar{h}L/k$ 为毕奥特(*Biot*)数，$\bar{h}$ 为全部曲面上对流的平均传热系数，k 为厚平板的热导。

在 $B_i=0.7$ 时求出 δ_n 的 20 个最低值，并用该值在 $0\leqslant\eta\leqslant1$ 及 $0\leqslant\tau\leqslant2$ 时画出曲面 $\theta(\eta,\tau)/\theta_i$。然后，用 rotate3d 函数交替得到一个可接受的曲面视图。标注轴并给出图形标题，加上垂直线以进一步加强曲面的可视化，如图 7.5 所示。

7.6　下式表示一外缘固定的方形薄膜，画出其形状及等高线图，图中在 $0\leqslant x\leqslant1$ 及 $0\leqslant y\leqslant1$ 时，每个方向使用 25 个网格点。

$$w_{23}(x,y)=\sin(2\pi x)\sin(3\pi y)$$

① 参见 D. R. Pitts 和 L. E. Sissom, *Theory and Practice of Heat Transfer*, Schaum's Outline Series, McGraw-Hill, New York, 1997年，第79页。

7.7 一个长为 l 的平板，流过平板的湍流的平均努塞尔特(Nusselt)数为：①

$$Nu=\frac{0.037Re^{0.8}Pr}{1+2.443Re^{-0.1}(Pr^{2/3}-1)}\quad 5\times10^5\leqslant Re\leqslant10^7\quad 0.6\leqslant Pr\leqslant 2\,000$$

其中 Re 为雷诺数，Pr 为普朗特(Prandtl)数。画出 $\log_{10}(Nu)$ 在所示范围内随 $\log_{10}(Re)$ 和 $\log_{10}(Pr)$ 变化的函数曲面。如图 7.5 所示，用垂线将图中平板的边界和曲面各角进行连接。

7.8 如图 7.23 所示，钢筋混凝土梁的中轴位置由参数 k 决定，k 值如下：②

$$k=-\rho n+\sqrt{(\rho n)^2+2\rho n}$$

其中 $\rho=A_s/bd$，$n=E_s/E_c$ 为钢筋和混凝土的弹性模量之比。画出 k 随 n 与 ρ 变化的曲面图，其中在 $6\leqslant n\leqslant12$ 内取 n 的 10 个值，在 $0.001\leqslant\rho\leqslant0.009$ 内取 ρ 的 9 个值，并在 $0.01\leqslant\rho\leqslant0.1$ 内取 ρ 的 10 个值。

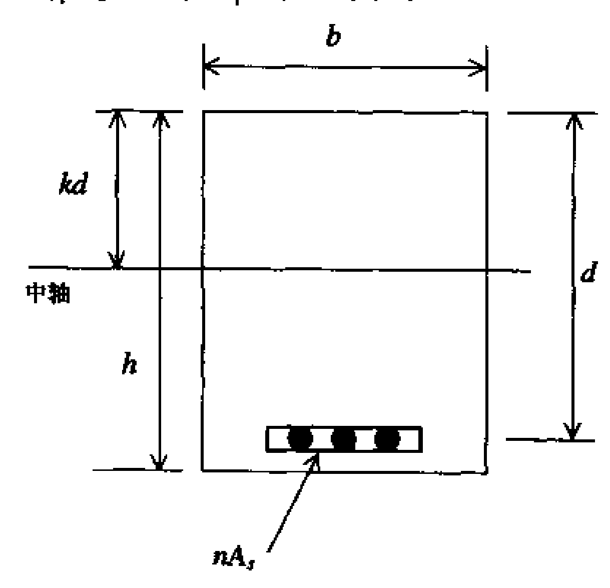


图 7.23 钢筋混凝土梁的剖面图

7.9 受扭力 T 作用的矩形梁的最大无量纲主剪切应力 τ' 由下式得到：③

$$\tau'^2=\tau'^2_{xz}+\tau'^2_{yz}$$

其中，

$$\tau'_{xz}=\frac{\tau_{xz}J}{Ta}=-\frac{16}{\pi^2}\sum_{n=0}^{\infty}\frac{(-1)^n}{(2n+1)^2}\frac{\sinh(k_n\xi)}{\cosh(k_nb/a)}\cos(k_n\eta)$$

$$\tau'_{yz}=\frac{\tau_{yz}J}{Ta}=2\eta-\frac{16}{\pi^2}\sum_{n=0}^{\infty}\frac{(-1)^n}{(2n+1)^2}\frac{\cosh(k_n\xi)}{\cosh(k_nb/a)}\sin(k_n\eta)$$

J 为转矩常数，$\eta=x/a(-1\leqslant\eta\leqslant1)$，$\xi=y/b(-1\leqslant\xi\leqslant1)$，$k_n=(2n+1)\pi/2$。

① 参见 W. Beitz 和 K. H. Kuttner 编辑，*Handbook of Mechanical Engineering*，Springer－Verlag，New York，1994 年，第 C31 页。

② 参见 L. Spiegel 和 G. F Limbrunner，*Reinforced Concrete Design*，第三版，Prentice Hall，Upper Saddle River，NJ，1992 年，第 196 页。

③ 参见 C. T. Wang，*Applied Elasticity*，McGraw－Hill，New York，1953 年，第 89 页。

在 $b/a=1$ 时，生成 $\tau'^2(\eta,\xi)$ 所表示的曲面，为方形横截面。在另一图上画出 $\tau'^2(\eta,\xi)$ 的等高线图，包括 30 条等高线。用 rotate3d 分析曲面视图，结果应如图 7.24 和图 7.25 所示。

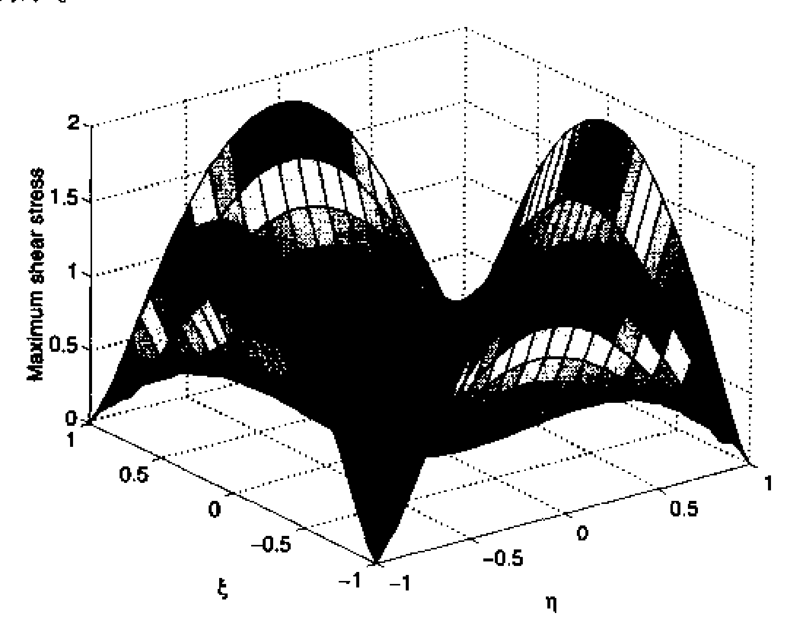


图 7.24 方形横截面梁所受扭力的最大剪切应力平方图

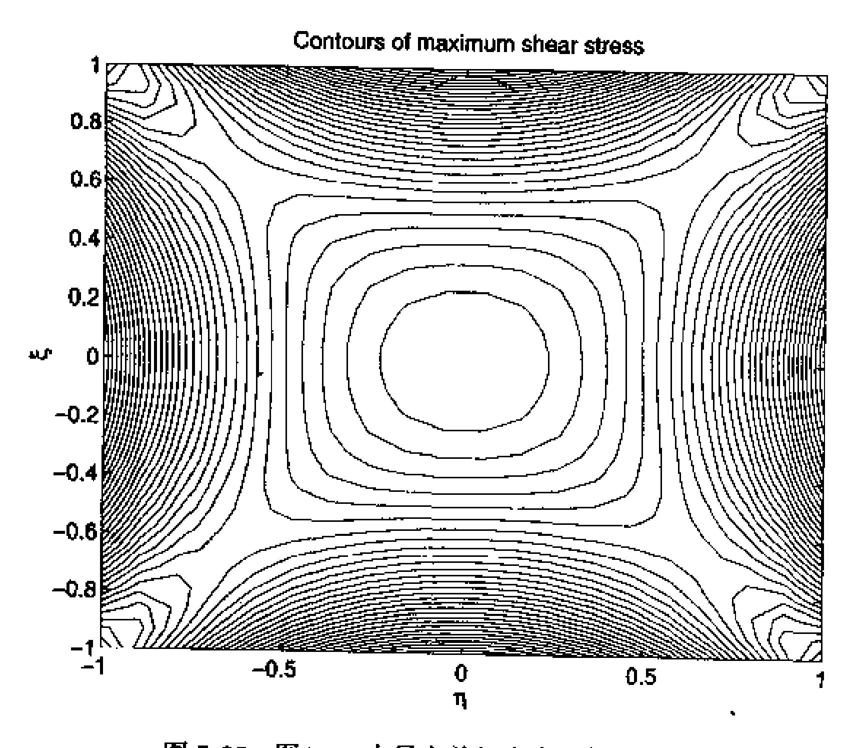


图 7.25 图 7.24 中最大剪切应力的等高线图

7.10 如表 7.1 所示，表中数据表明当某过程输入量为 x_1 和 x_2 时，输出量是相对于其均值的偏差。使用 stem3 函数绘制函数图形，得到图 7.26，要求必须使用单独命令画出 $z=0$ 处的平面。使用 view（−30，7）得到图中所示方向（二维函数为 stem）。



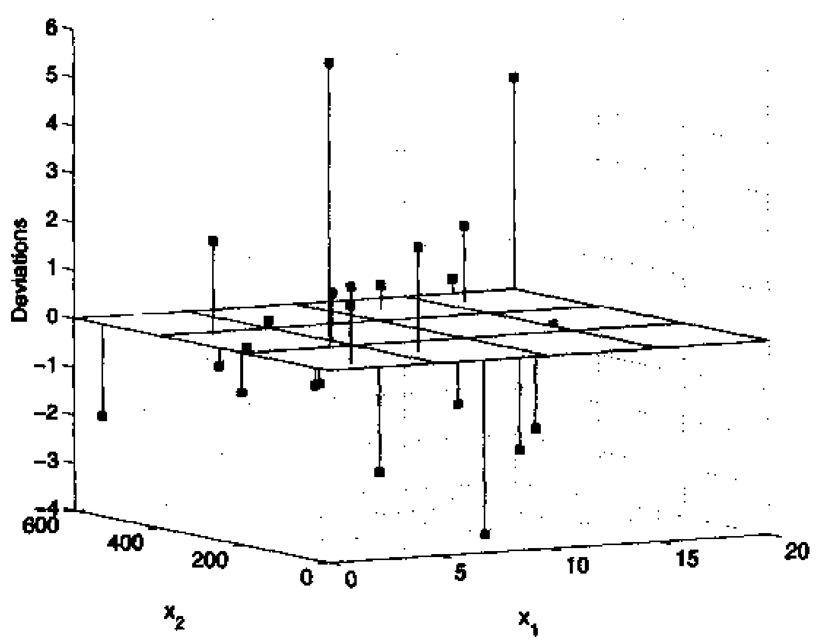


图 7.26　用函数 stem3 得到的以平面 $z=0$ 为基准的偏差

表 7.1　过程标准偏差

x_1	x_2	z	x_1	x_2	z
2	50	1.571 3	2	360	−0.602 3
8	110	−1.146 0	4	205	5.840 9
11	120	−2.204 1	4	400	−0.362 0
10	550	−1.596 8	20	600	4.334 1
8	295	−2.893 7	1	585	−2.036 8
4	200	1.113 6	10	540	−1.541 5
2	375	1.929 7	15	250	0.030 2
2	52	1.196 2	15	290	−2.180 9
9	100	−3.865 0	16	510	1.558 7
8	300	−0.476 3	17	590	0.322 2
4	412	−1.322 3	6	100	2.147 8
11	400	−0.461 9	5	400	0.153 7
12	500	0.491 1			

第 8 章　机械零件设计

本章介绍不同类型机械零件的各种分析方法。

8.1　向量、受力和刚体平衡

研究向量：

$$\boldsymbol{a} = a_1\boldsymbol{i} + a_2\boldsymbol{j} + a_3\boldsymbol{k}$$

点乘定义为：

$$\boldsymbol{a}\cdot\boldsymbol{a} = a_1^2 + a_2^2 + a_3^2 \tag{8.1}$$

其值由下式获得：

$$a = |\boldsymbol{a}| = \sqrt{\boldsymbol{a}\cdot\boldsymbol{a}} = \sqrt{{a_1}^2 + {a_2}^2 + {a_3}^2} \tag{8.2}$$

如果 $\boldsymbol{a}$ 如下表示：

```
a = [a1 a2 a3]
```

那么点乘由下面的函数表示：

```
adot = dot(a,a)
```

其值可以是：

```
maga = sqrt(dot(a,a))
```

或者是：

```
maga = norm(a)
```

$\boldsymbol{a}$ 的方向余弦是：

$$\cos(\alpha_j) = \frac{a_j}{|\boldsymbol{a}|}\quad j = 1,2,3 \tag{8.3}$$

上式为单位向量 $\boldsymbol{u}_a$ 在 $\boldsymbol{a}$ 方向上的分量，可以写为：

$$\boldsymbol{u}_a = \cos\alpha_1\boldsymbol{i} + \cos\alpha_2\boldsymbol{j} + \cos\alpha_3\boldsymbol{k} = \frac{a_1}{|\boldsymbol{a}|}\boldsymbol{i} + \frac{a_2}{|\boldsymbol{a}|}\boldsymbol{j} + \frac{a_3}{|\boldsymbol{a}|}\boldsymbol{k} \tag{8.4}$$

所以，

$$\boldsymbol{a} = |\boldsymbol{a}|\boldsymbol{u}_a \tag{8.5}$$

并且，

$$\boldsymbol{u}_a\cdot\boldsymbol{u}_a = \cos^2\alpha_1 + \cos^2\alpha_2 + \cos^2\alpha_3 = \frac{a_1^2}{|\boldsymbol{a}|^2} + \frac{a_2^2}{|\boldsymbol{a}|^2} + \frac{a_3^2}{|\boldsymbol{a}|^2} = 1$$

其中,

$$\alpha_j = \arccos \frac{a_j}{|\boldsymbol{a}|} \quad j = 1,2,3 \tag{8.6}$$

在 $\boldsymbol{a}$ 方向上的单位向量 $\boldsymbol{u}_a$ 可如下表达:

```
ua = a/norm(a)
```

方向余弦角度的弧度值是:

```
alpha = acos(a/norm(a))
```

角度值是:

```
alphadeg = acos(a/norm(a)) * 180/pi
```

若有另一个向量:

$$\boldsymbol{b} = b_1\boldsymbol{i} + b_2\boldsymbol{j} + b_3\boldsymbol{k}$$

那么,如果 $\boldsymbol{c} = \boldsymbol{a} \pm \boldsymbol{b}$,则:

$$\boldsymbol{c} = (a_1 \pm b_1)\boldsymbol{i} + (a_2 \pm b_2)\boldsymbol{j} + (a_3 \pm b_3)\boldsymbol{k} \tag{8.7}$$

向量的加减分别是:

```
c = a + b
```

和

```
c = a - b
```

向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 的叉乘被定义为:

$$\boldsymbol{c} = \boldsymbol{a} \times \boldsymbol{b} = \begin{vmatrix} \boldsymbol{i} & \boldsymbol{j} & \boldsymbol{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix} = (a_2 b_3 - a_3 b_2)\boldsymbol{i} + (a_1 b_3 - a_3 b_1)\boldsymbol{j} + (a_1 b_2 - a_2 b_1)\boldsymbol{k} \tag{8.8}$$

其中 c 垂直于包含 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 的平面。叉乘表示为:

```
c = cross(a,b)
```

为了在单位向量 $\boldsymbol{u}_a$ 的方向上确定叉乘的值,使用 3 级标量乘积:

$$c_a = \boldsymbol{u}_a \cdot \boldsymbol{a} \times \boldsymbol{b} \tag{8.9}$$

其中 C_a 是一个标量,那么,

```
ca = dot(ua.cross(a,b))
```

这些结论可用来分析刚体部件的受力和力矩等一系列问题。

例 8.1 力的合成

一个力系如图 8.1 所示。确定合力 $\boldsymbol{F}$ 的值及其方向余弦角 α_i 的大小。

$$\boldsymbol{F}_R = \boldsymbol{F}_1 + \boldsymbol{F}_2$$

由式(8.6)得:

$$\alpha_i = \arccos \frac{F_{Ri}}{|\boldsymbol{F}_R|}$$

程序清单如下：

```
F1=[0 40 60];F2=[60 -110 70];
resultant=norm(F2+F1)
angles=acos((F2+F1)/resultant)*180/pi
```

执行程序，得到 $|F| = resultant = 159.3738$，$\alpha = angles = [67.8846, 116.0541, 35.3441]$，用角度表示。

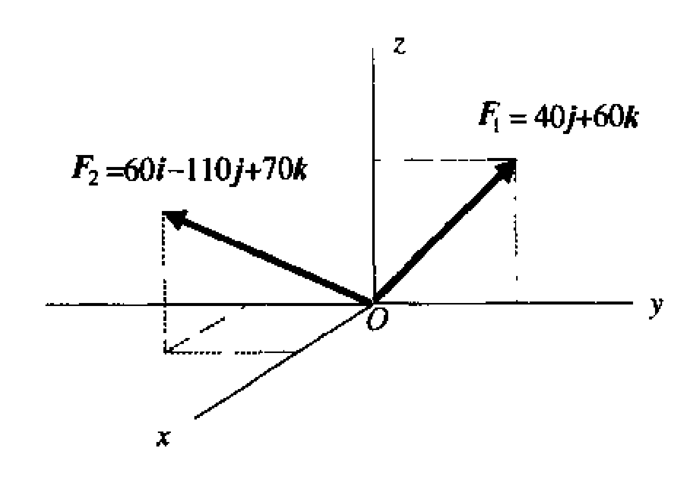


图 8.1　例 8.1 的受力方向

例 8.2　力的分解

一个力如图 8.2 所示。使用式(8.4)和式(8.5)分解作用在 C 点的力 F_C，并用式(8.6)确定该力方向余弦角 α_C。程序清单如下：

```
r=[24-16-48]
ur=r/norm(r);
Fcomps=30*ur
angles=acos(ur)*180/pi
```

执行程序后得到 $F_C = Fcomps = [12.8571, -8.5714, -25.7143]$，$\alpha_C = angles = [64.6231, 106.6015, 148.9973]$，用角度表示。

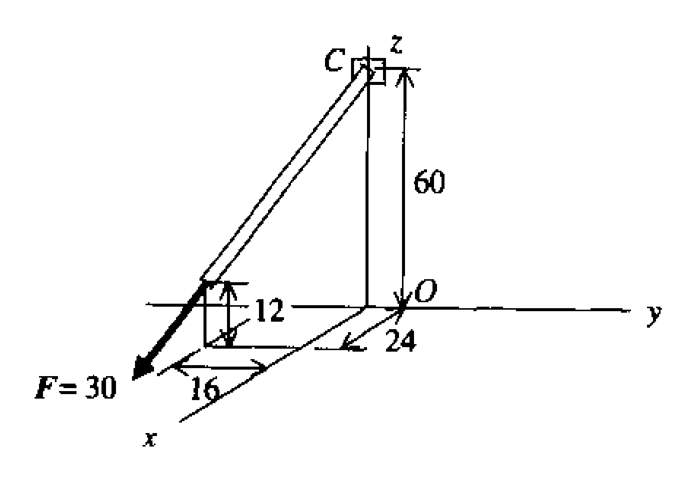


图 8.2　例 8.2 的受力方向

例 8.3 合力的值

一个力系如图 8.3 所示。求 F_L 和 F_R 的合力大小及其分量。

$$F = F_L + F_R$$

使用式(8.2)、式(8.4)及式(8.5)程序清单如下：

```
rl = [2.5   0 -3]; rr = [2.5   0.5 -3];
F = 35 * rl/norm(rl) + 25 * rr/norm(rr)
resultant = norm(F)
```

程序执行后得到 $|F|$ = *resultant* = 59.881 8。F 分量是 F = [38.281 5, 3.175 0, -45.937 8]。

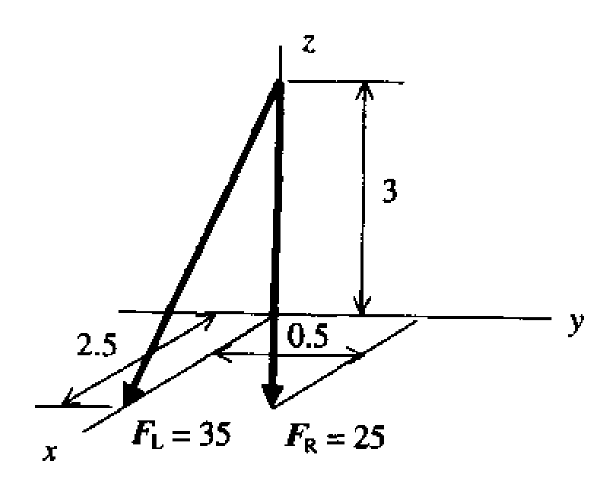


图 8.3 例 8.3 的受力方向

例 8.4 给定方向上的受力分解

分析如图 8.4 所示的力，确定 F 相对于构件 OA 平行方向上的分量(F_{OA})和垂直方向上的分量($F_{\perp}$)，并确定 $F_{\perp}$。使用式(8.4)、(8.5)和式(8.2)，并记为：

$$F = F_{OA} + F_{\perp}$$

程序清单是：

```
r = [1   3   1.5]; F = [0   125   0];
ur = r/norm(r);
FoaPar = dot(F, ur) * ur
FoaPerp = F-FoaPar
FoaPerpMag = norm(FoaPerp)
```

执行程序后得到：F_{OA} = *FoaPar* = [30.612 2, 91.836 7, 45.918 4]，$F_{\perp}$ = *FoaPerp* = [-30.612 2, 33.163 3, -45.918 4]，$|F_{\perp}|$ = *FoaPerpMag* = 64.384 8。

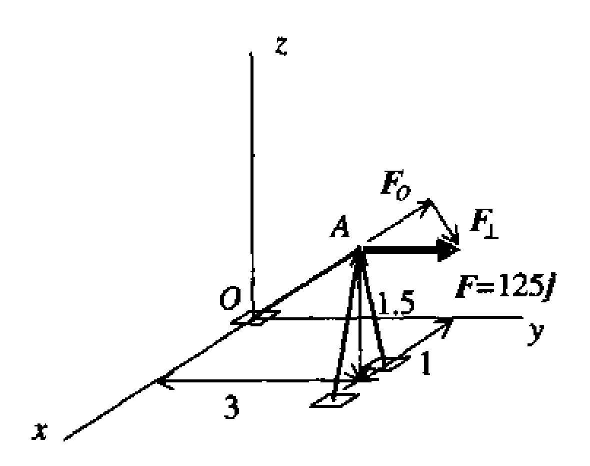


图 8.4　例 8.4 的受力方向

例 8.5　力的平衡

一个力系如图 8.5 所示。确定保持该力系平衡所需的力的分量、大小和方向余弦角 α。

$$\boldsymbol{F} = -(\boldsymbol{F}_1 + \boldsymbol{F}_2 + \boldsymbol{F}_3 + \boldsymbol{F}_4)$$

使用式(8.2)、式(8.4)、式(8.5)和式(8.6)程序清单如下：

```
F1 = [0 275 0];F2 = [0 0 - 150];
F4 = [ - 100 0 0];r = [ - 2  - 6 7];
F3 = 400 * r/norm(r);
Fequil = - (F1 + F2 + F3 + F4)
FequilMag = norm(Fequil)
angles = acos(Fequil/FequilMag) * 180/pi
```

程序执行后得到：$\boldsymbol{F}$ = *Fequil* = [184.799 8, − 20.600 5, − 146.799 4]，|$\boldsymbol{F}$| = *FequilMag* = 236.908 0，α = *angles* = [38.735 0,94.988 5,128.290 4]，用角度表示。

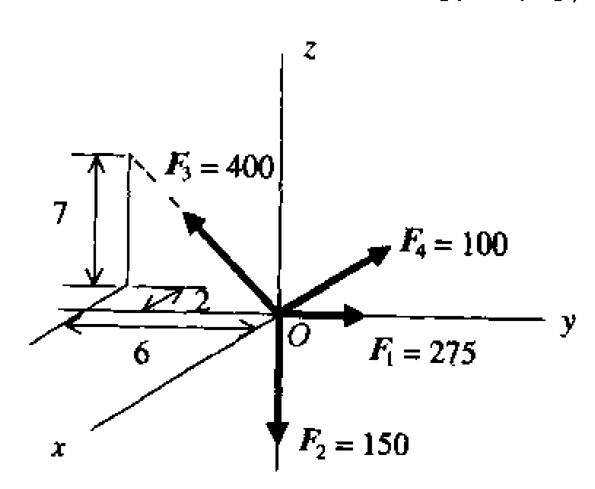


图 8.5　例 8.5 的受力方向

例 8.6　力的大小

一个力系如图 8.6 所示。当 $\boldsymbol{W} = -85\boldsymbol{k}$ 时，确定 F_1、F_2 和 F_3 的值。首先确定分力 $\boldsymbol{F}_j = |\boldsymbol{F}_j|\boldsymbol{u}_j$，其中 $\boldsymbol{u}_j$ 是在 $\boldsymbol{r}_j = r_{j1}\boldsymbol{i} + r_{j2}\boldsymbol{j} + r_{j3}\boldsymbol{k}$ 方向上的单位向量，$j = 1,2,3$，从式(8.4)和式(8.5)中确定，有下式：

$$F_1 + F_2 + F_3 + W = 0$$

或

$$H_1 i + H_2 j + H_3 k = 85k$$

$$H_n = \sum_{m=1}^{3} |F_m/r_m| r_{mn} \qquad n = 1,2,3$$

写成矩阵形式如下：

$$\begin{bmatrix} r_{11}/|r_1| & r_{21}/|r_2| & r_{31}/|r_3| \\ r_{12}/|r_1| & r_{22}/|r_2| & r_{32}/|r_3| \\ r_{13}/|r_1| & r_{23}/|r_2| & r_{33}/|r_3| \end{bmatrix} \begin{bmatrix} |F_1| \\ |F_2| \\ |F_3| \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 85 \end{bmatrix}$$

程序清单是：

```
r1 = [1 0 0];r2 = [ -4  -3 7];r3 = [ -4 2 7];
u1 = r1/norm(r1);
u2 = r2/norm(r2);
u3 = r3/norm(r3);
Fmag 123 = [u1' u2' u3'] \ [0 0 85]'
```

程序执行后得到$|F_j|$ = *Fmag123* = [48.571 4 41.782 7,60.519 7]′；即$|F_1|$ = *Fmag123*(1) = 48.571 4，$|F_2|$ = *Fmag123*(2) = 41.782 7 和$|F_3|$ = *Fmag123*(3) = 60.519 7。

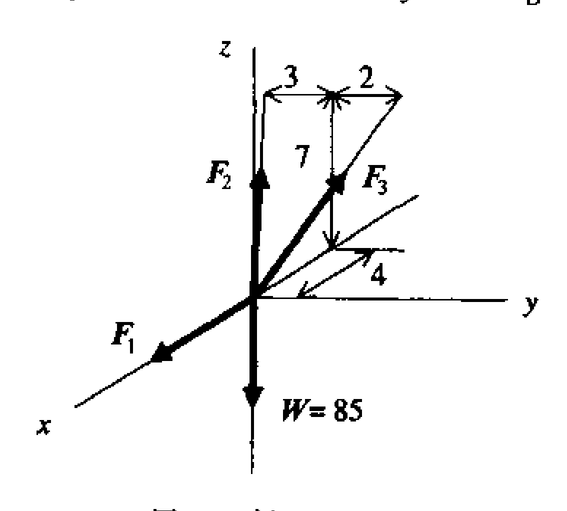


图 8.6　例 8.6 的受力方向

例 8.7　力矩计算

受力如图 8.7 所示。确定 F 作用在构件 OA 上对 O 点的力矩大小。如果 r_a 表示从 O 到 A 的位置向量，r_b 表示从 O 到 B 的位置向量，则 F 在 BA 方向上的分量为 $|F| u_f$，其中，u_f 是 $r_a - r_b$ 的单位向量，那么由式(8.8)可得出力矩是：

$$M = r_a \times (|F| u_f)$$

并且它的值可以由式(8.2)得到，程序清单如下：

```
ra = [1.5 3.5 3];rb = [3.5 5 0];
F = 25 * (ra - rb)/norm(ra - rb);
FaboutA = cross(ra,F);
Mmag = norm(FaboutA)
```

程序执行后得到：$|M| = Mmag = 121.096\,8$。

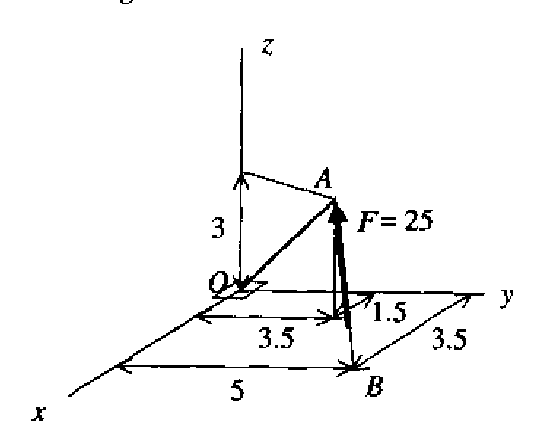


图 8.7　例 8.7 的受力方向

例 8.8　多个力作用下的力矩

作用在一个构件 OAB 上的力系如图 8.8 所示。确定在 O 点由 $\boldsymbol{F}_1,\boldsymbol{F}_2,\boldsymbol{F}_3$ 引起的力矩 M 的值以及方向余弦角 α 的值。这样，

$$\boldsymbol{M} = \boldsymbol{r}_a \times \boldsymbol{F}_1 + \boldsymbol{r}_a \times \boldsymbol{F}_2 + \boldsymbol{r}_b \times \boldsymbol{F}_3$$

其中 $\boldsymbol{r}_a$ 是位置向量 OA，$\boldsymbol{r}_b$ 是位置向量 OB，程序清单如下：

```
ra = [0 11 0]; rb = [9 11 0];
F1 = [ - 55 45 30]; F2 = [0 20 0]; F3 = [70 40 - 50];
Mro = cross(ra,F1) + cross(ra,F2) + cross(rb,F3);
MroMag = norm(Mro)
angles = acos(Mro/MroMag) * 180/pi
```

程序执行后得到：$|M| = MroMag = 537.517\,4$，$\alpha = angles = [114.160\,2, 33.156\,3, 68.729\,0]$，用角度表示。

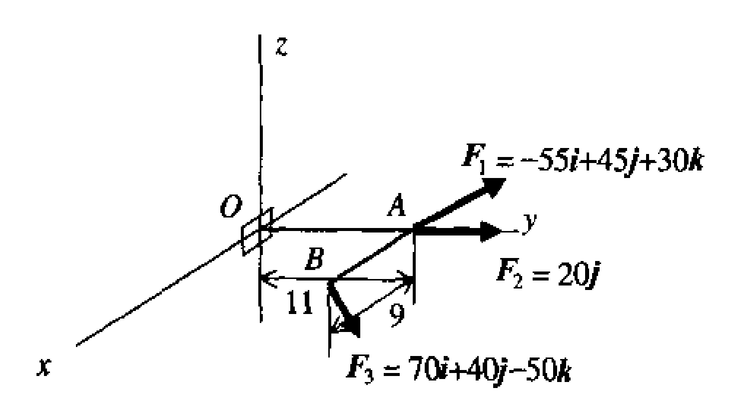


图 8.8　例 8.8 的受力方向

例 8.9　绕轴转矩

一个力系如图 8.9 所示，其中：

$$\boldsymbol{F} = -30\boldsymbol{i} + 10\boldsymbol{j} - 15\boldsymbol{k}$$

是作用在构件 ABC 上的力，确定力 $\boldsymbol{F}$ 引起的构件 AB 绕轴转矩。力矩表示为：

$$|M_{AB}| = \boldsymbol{u}_b \cdot \boldsymbol{r}_d \times \boldsymbol{F}$$

和

$$M_{AB} = |M_{AB}| u_b$$

其中 u_b 是 AB 方向上的单位向量，r_d 是 AD 方向上的向量，程序清单如下：

```
rb=[0 0 1.5];rd=[0 1 0];F=[-30 10 -15];
ub=rb/norm(rb);
Mab=dot(ub,cross(rd,F))
MabVec=Mab*ub
```

执行程序后得到：$|M_{AB}| = Mab = 30$，$M_{AB} = MabVec = [0\ 0\ 30]$。

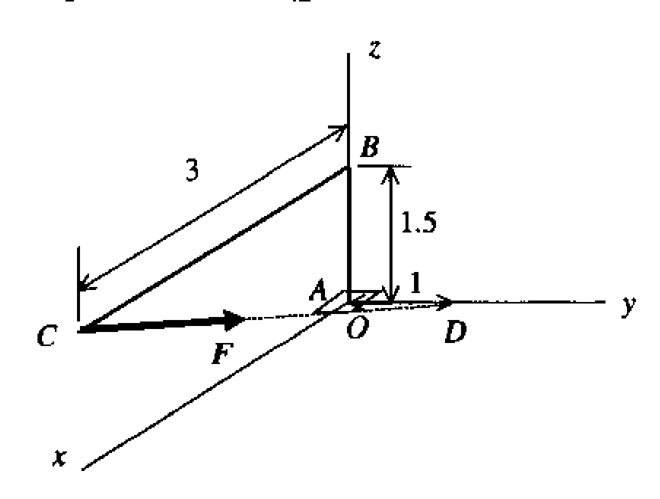


图 8.9 例 8.9 的受力方向

例 8.10 等效力和力矩

一个力系如图 8.10 所示，其中 M 是关于图示轴线的力矩，通过作用在 O 点的等效合力 F_o 和力偶 M_o 代替力系的作用，这样，如果 r_{AF} 是一个表示从 A 到 M 方向上的位置向量（$\alpha = 36.87°$ 表示直角三角形的一个角）并且 $r_{21} = r_2 - r_1$，那么：

$$F_2 = F_2 r_{21}/|r_{21}|$$

$$M = M r_{AF}/|r_{AF}|$$

$$F_o = F_1 + F_2$$

和

$$M_O = M + r_1 \times F_1 + r_2 \times F_2$$

程序清单如下：

```
r1=[0 0 12.5];r2=[-1.75 1.25 12.5];
rAF=[0 -4 3];F1=[0 0 -40];
r21=r2-r1;
F2=15*r21/norm(r21);
M=25*rAF/norm(rAF);
FatO=F1+F2
MatO=M+cross(r1,F1)+cross(r2,F2)
```

程序执行后得到：$F_O = FatO = [-12.2060, 8.7186, -40.0000]$，$M_O = MatO = [-108.9822, -172.5750, 15.0000]$。

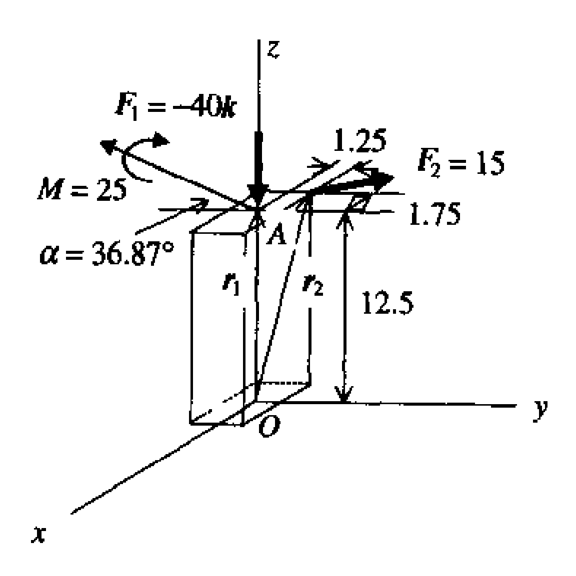


图 8.10　例 8.10 的受力方向

8.2 梁、压杆、轴的应力和偏差

8.2.1 静定梁

如果定截面梁横向位移为 $w(x)$，长为 L，弹性模量是 E，横截面的惯性矩是 I，单位长度载荷是 $P_o q(x)$，则无量纲方程如下：

$$\frac{d^4 y}{d\eta^4} = q(\eta) \tag{8.10}$$

其中 $\eta = x/L, y = y(\eta) = w/h_o, h_o = P_o L^4/EI$，$P_o$ 是常数，表示 q 的最大值。倾角 θ_d，中轴的弯矩 M_d 和剪应力 V_d 用无量纲位移 y 表示为：

$$\begin{aligned} \theta &= \frac{\theta_d}{P_o L^3/EI} = \frac{dy}{d\eta} \\ M &= \frac{M_d}{P_o L^2} = \frac{d^2 y}{d\eta^2} \\ V &= \frac{V_d}{P_o L} = \frac{d^3 y}{d\eta^3} \end{aligned} \tag{8.11}$$

使用无量纲量能得到数值解。这些方法明显地与梁的几何形状、物理特性和载荷的大小无关，而与边界条件和梁的载荷形式有关。

梁中最大的剪应力和弯曲应力分别是：

$$\tau = \frac{\alpha V}{A} \text{和} \sigma = \frac{Mc}{I}$$

其中 A 是梁的横截面的面积，α 是一个关于梁的横截面形状的系数，c 是中性轴到梁横截面顶端（如果不对称，则为底端，）边缘的距离。横截面是实心圆时，$\alpha = 4/3$；是矩形时，$\alpha = 3/2$；是圆环时，$\alpha = 2$。

使用函数 ode45 和 fsolve 来对不同边界条件与载荷情况下的问题求解，是一种数值解法，并不能给出像式(8.10)那样的解析式。这些已经在第 5.5.4 节和第 5.5.5 节中分别进行过讨

论。

为了把式(8.10)转换成 ode45 允许的形式，将式(8.10)重新表示为 4 个一阶方程组，即：

$$y_1 = y \qquad y_2 = \frac{dy}{d\eta}$$
$$y_3 = \frac{d^2y}{d\eta^2} \qquad y_4 = \frac{d^3y}{d\eta^3}$$

式(8.10)变成：

$$\frac{dy_1}{d\eta} = y_2 \qquad \frac{dy_2}{d\eta} = y_3$$
$$\frac{dy_3}{d\eta} = y_4 \qquad \frac{dy_4}{d\eta} = q \tag{8.12}$$

参见第 5.5.4 节，ode45 的输出是一个表示 n 个 η 值的行向量和一个($n\times4$)维数组(记为 yy)。其中：

$$yy(:,1) = y \qquad yy(:,3) = d^2y/d\eta^2 = M$$
$$yy(:,2) = dy/d\eta = \theta \quad yy(:,4) = d^3y/d\eta^3 = V$$

在 yy 中 $\eta=0$ 和 $\eta=1$ 对应梁的每一端的边界条件，三种最常用梁的边界条件由表 8.1 给出。

表 8.1 梁的通用边界条件及其与 ode45 的关系

		From ode45[①]	
类型	需求	$\eta=0$	$\eta=1$
简支端($y=M=0$)	$y=d^2y/d\eta^2=0$	$yy(1,1)=yy(1,3)=0$	$yy(end,1)=yy(end,3)=0$
固定端($y=\theta=0$)	$y=dy/d\eta=0$	$yy(1,1)=yy(1,2)=0$	$yy(end,1)=yy(end,2)=0$
自由端($M=V=0$)	$d^2y/d\eta^2=d^3y/d\eta^3=0$	$yy(1,3)=yy(1,4)=0$	$yy(end,3)=yy(end,4)=0$

① $yy(:,1)=y$；$yy(:,2)=dy/d\eta=\theta$；$yy(:,3)=d^2y/d\eta^2=M$；$yy(:,4)=d^3y/d\eta^3=V$。

使用 ode45 仅能确定 $\eta=0$ 的边界条件。对于四阶方程要确定四个边界条件。这样在 $\eta=0$处得到两个，在 $\eta=1$ 处得到两个。问题是如何用 $\eta=0$ 的两个未知边界条件去求 $\eta=1$ 的边界条件。利用函数 fsolve 可进行数值化近似求解，此函数可用来求解非线性方程 $F(x)=0$。在这种情况下，用 $F(x)$代表 $\eta=1$ 处由 ode45 求出的式(8.12)的两个边界条件。在求解过程中，两个独立参数是 $\eta=0$ 处未知的边界条件。

求解过程如下。假设 $\eta=0$ 处两个不确定的边界条件，使用 ode45 求解式(8.12)，看 $\eta=1$ 时的边界条件是否得到满足。如果不满足，再用 fsolve 选择另一组在 $\eta=0$ 时的不确定边界条件值，ode45 又得到式(8.12)的一个解。这一过程重复执行，直到得到一个可以接受的 $\eta=1$ 时的边界条件近似解为止。

现在用一系列的例子来说明。首先创建一些函数提供灵活性。创建两个明显不同的载荷分布函数：一个是集中载荷函数，另一个是分布载荷函数。参见图 8.11，假设分布载荷是：

$$q(\eta) = \left[\alpha + \beta\left(\frac{\eta - m}{1 - n - m}\right)^k\right][u(\eta - m) - u(\eta - 1 + n)] \tag{8.13}$$

其中，$u(x)$是一个单位阶跃函数，$k>0,1\geqslant n+m\geqslant 0,m\geqslant 0,n\geqslant 0$，$\alpha=0$ 或 1，并且 $\beta\geqslant 0$。在 $\eta=\zeta,0\leqslant\zeta\leqslant 1$ 时，集中载荷为：

$$q(\eta)=1/2\varepsilon \qquad \zeta-\varepsilon<\eta<\zeta+\varepsilon \tag{8.14}$$

其中 2ε 是在 ode45 中确定的 η 相邻值的增量。

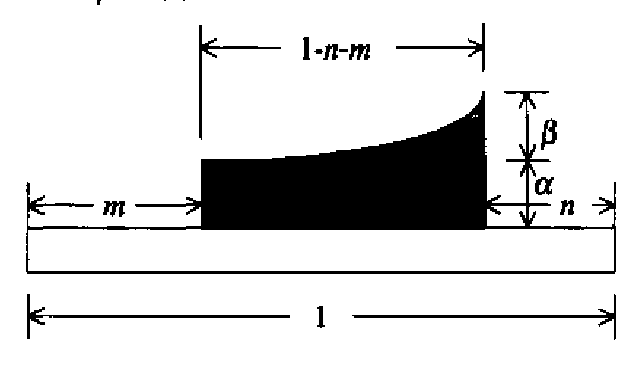


图 8.11　梁上普通载荷的术语定义

如果调用函数 *Beamload* 计算载荷，则计算式(8.13)和式(8.14)的程序如下：

```
function lod = BeamLoad(eta,alpha,m,n,k,beta,xi,flag)
lod = 0;
if flag = = 0
  if (eta > xi - .012)&(eta < xi + .012)
   lod = 1;
  end
else
  if (eta > = m)&(eta < = 1 - n)
   lod = alpha + beta * ((eta-m)/(1 - n - m))^k;
  end
end
```

其中，$eta=\eta$，$alpha=\alpha$，$m=m$，$n=n$，$k=k$，$beta=\beta$，$xi=\zeta$ 并且 *flag* = 0 或 1。当 *flag* = 0 时，得到集中载荷；当 *flag* = 1 时，得到分布载荷。η 的增量为 0.02，若令 $\varepsilon=0.012$，则 $2\varepsilon=0.024$，略大于 0.02。

ode45 需要一个表示式(8.12)给出的四个一阶方程的列向量函数。在第 5.5.4 节中，曾调用过函数 *StaticBeam*，语句如下：

```
function sb = StaticBeam(eta,w,d,alpha,m,n,k,beta,xi,flag)
sb = [w(2);w(3);w(4);BeamLoad(eta,alpha,m,n,k,beta,xi,flag)];
```

其中，$w(2)=y_2$，$w(3)=y_3$，$w(4)=y_4$，$BeamLoad=q$，均由式(8.13)和式(8.14)给出，d 是 ode45 需要的，但在这种情况下没有用到。

通过使用函数 *BeamPlot* 来绘出和显示数字结果，这个函数应该是：

- 绘制沿梁的位移、转角、力矩和剪应力；
- 显示梁每一端的剪应力和转角值；
- 显示位移、转角、力矩和剪应力的最大值及其位置坐标。

该函数要处理如下两种不同的情况：①

1. 当 *over* = 0 时，应满足本节给出的静定梁的情况。
2. 当 *over* = 1 时，应满足第 8.2.2 节讨论的外伸梁的情况。

变量 *lenover* 是梁的悬臂部分的长度，变量 *rho*、*alpha_over*、*m_over*、*n_over*、*k_over*、*beta_over*、*xi_over* 和 *flag_over* 分别表示悬臂部分的载荷，变量 *eta*、*alpha*、*m*、*n*、*k*、*beta*、*xi* 和 *flag* 分别表示梁的参数。*over* = 0 时，*ic* 是四元向量，表示 $\eta = 0$ 时的四个边界条件，其中两个给定，另两个由 fsolve 函数确定。

研究一个左端简单支撑的外伸梁。当 *over* ≠ 0 时，*ic* 的前两个元素分别是 $\eta = 0$ 处，由 fsolve 求出的倾角和剪应力。后两个元素是在 $\rho = 0(\eta = 1$，见图 8.16)时外伸梁左端部分的力矩和剪应力，这些也由 fsolve 函数确定，值 *lbc* 和 *rbc* 是分别用来描述梁左右两端边界条件的字符串，在外伸梁中 *rbc* 对应于梁的右端悬臂部分。最后，用标尺 100 和 $den = 2\varepsilon = 0.024$ 绘出结果，程序如下：

```
function BeamPlot (alpha,m,n,k,beta,xi,flag,ic,lbc,rbc,over,lenover,…
                  alpha_over,m_over,n_over,k_over,beta_over,xi_over,…
                  flag_over)
if over= =0
  [eta,y] = ode45('StaticBeam',[0:0.02:1],ic',[],alpha,m,n,k,beta,xi,flag);
else
  [eta,y] = ode45('StaticBeam',[0:0.02:1],[0 ic (1) 0 ic (2)]',[],alpha,m,n,k,beta,xi,flag);
  [zeta,z] = ode45 ('StaticBeam',[0:0.02:lenover],[0 y(end,2)ic (3) ic (4)]',[],…
                  alpha_over,m_over,n_over,k_over,beta_over,xi_over,flag_over);
  eta = [eta;1 + zeta];
  y = [y;z];
end
y = - y;den = 1;
if flag= =0,den = 0.024;end
subplot(2,2,1)
plot(eta,y(:,1) * 100/den,'k - ',[0 1 + lenover],[0 0],'k - - ')
if over= =0
  if flag= =0
  title (['Point load at \ xi = 'num2str(xi,2)])
else
  title (['Load: \ alpha = 'num2str(alpha,2)'m = 'num2str(m,2)'n = '…
        num2str(n,2)'k = 'num2str(k,2)' \ beta = 'num2str(beta,2)])
end
ylabel('displacement( \ times10^{ - 2})')
xlabel(' \ eta')
[ymax,index] = max(abs(y(:,1))/den);
disp(['max displacement = 'num2str(ymax,3)'at eta = 'num2str(eta(index),3)])
subplot(2,2,2)
plot(eta,y(:,2)/den,'k - ',[0 1 + lenover],[0 0],'k - - ')
ylabel('slope')
xlabel(' \ eta')
```

① 该绘图函数用于绘制本节和第 8.2.2 节外伸梁的计算结果。over = 1 的情况将在以后章节中讨论。

```
title(['left bc:'lbc' right bc:'rbc])
[tmax,index]=max(abs(y(:2))/den);
disp('slope:')
disp([''num2str (y(1,2)/den,3)'at eta=0'])
disp([''num2str (y(end,2)/den,3)'at eta=1'])
disp(['max slope='num2str(tmax,3)'at eta='num2str(eta(index),3)])
subplot(2,2,3)
plot(eta,y(:,3)/den,'k-',[0 1+lenover],[0 0],'k--')
ylabel('moment')
xlabel('\eta')
[mmax,index]=max(abs(y(:3))/den);
disp(['max monment='num2str(mmax,3)'at eta='num2str(eta(index),3)])
subplot(2,2,4)
plot(eta,y(:,4)/den,'k-',[0 1+lenover],[0 0],'k--')
ylabel('shear')
xlabel('\eta')
[vmax,index]=max(abs(y(:4))/den);
disp('shear:')
disp([''num2str(y(1,4)/den,3)'at eta=0'])
disp([''num2str(y(end,4)/den,3)'at eta=1'])
disp(['max shear='num2str(vmax,3)'at eta'='num2str(eta(index),3)])
```

通过下例说明求解过程。

例 8.11　均布载荷作用的简支梁

分析一个均布载荷作用的简支梁。边界条件在表 8.1 中的第一行给出，并且该载荷用下列参数描述：$\alpha=1, m=n=k=\beta=\zeta=0$ 和 $flag=1$，程序清单如下：

```
alpha=1,m=0;n=0;k=0;beta=0;xi=0;flag=1;
alpha_over=0;m_over=0;n_over=0;k_over=0;
beta_over=0;xi_over=0;flag_over=0;
over=0;lenover=0;
options=optimset('display','off');
x=fsolve('BeamSolveSSSS',[.2 .5],options,alpha,m,n,k,beta,xi,flag);
BeamPlot (alpha,m,n,k,beta,xi,flag,[0 x(1) 0 x(2)]),'ss','ss',over,lenover,...
        alpha_over,m_over,n_over,k_over,beta_over,xi_over,flag_over);
```

其中函数 *BeamSolveSSSS* 是：

```
function d=BeamSolveSSSS(bc,alpha,m,n,k,beta,xi,flag)
[eta,y]=ode45('StaticBeam',[0:02:1],[0 bc(1) 0 bc(2)]',[],alpha,m,n,k,beta,xi,flag);
d=[y(end,1);y(end,3)];
```

执行程序后结果如图 8.12 所示，在 MATLAB 命令窗口中显示下面的信息：

```
max displacement=0.013 at eta=0.5
slope:
  -0.0417 at eta=0
  0.0417 at eta=1
max slope=0.0417 at eta=1
max moment=0.125 at eta=0.5
shear:
0.5 at eta=0
```

```
- 0.5 at eta = 1
max shear = 0.5 at eta = 0
```

该问题采用解析方法① 得到和上面相同的结果。

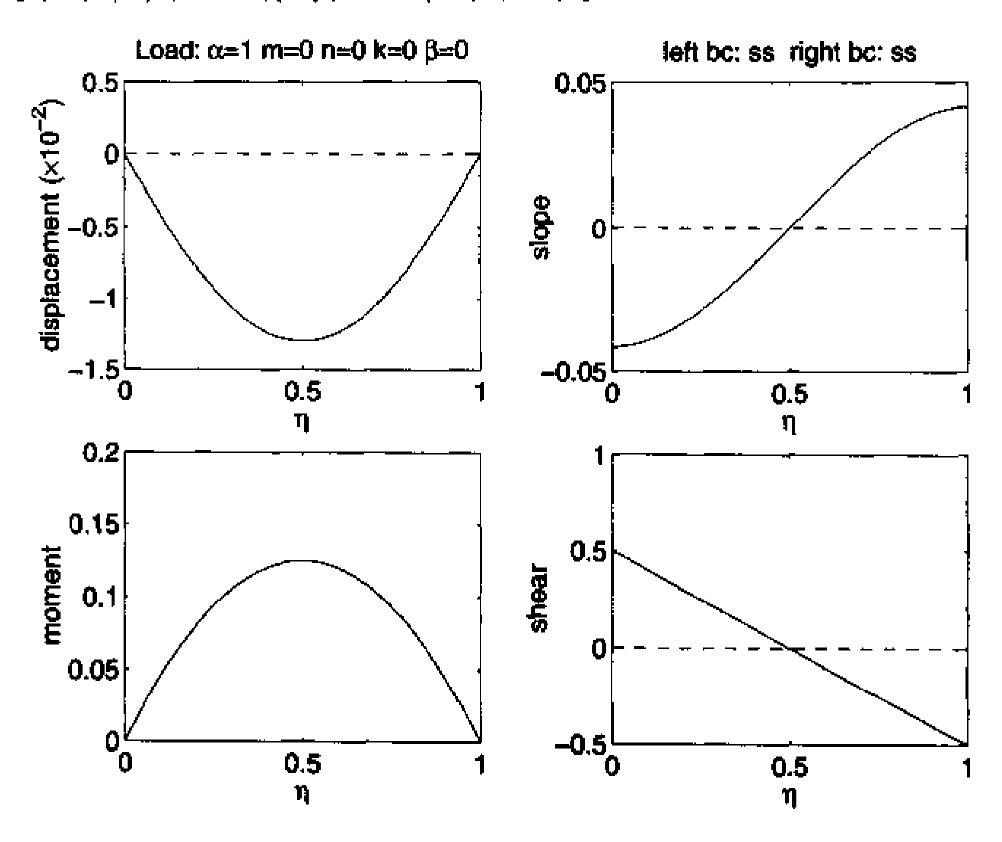


图 8.12 沿长度均布载荷作用的简支梁

例 8.12 集中载荷作用的简支梁

研究一个两端支撑的简支梁，在 $\eta=0.5$ 处受集中载荷作用。边界条件在表 8.1 的第一行给出，并且载荷由下列参数描述：$\alpha=m=n=k=\beta=0$，$\zeta=0.5$ 和 $flag=0$。程序清单如下：

```
alpha = 1;m = 0;n = 0;k = 0;beta = 0;xi = 0.5;flag = 0;
alpha_over = 0;m_over = 0;n_over = 0;k_over = 0;
beta_over = 0;xi_over = 0;flag_over = 0;
over = 0;lenover = 0;
options = optimset('display','off');
x = fsolve('BeamSolveCant',[.2.5],options,alpha,m,n,k,beta,xi,flag);
BeamPlot(alpha,m,n,k,beta,xi,flag,[0x(1)x(2)],'ss','ss',over,lenover,...
        alpha_over,m_over,n_over,k_over,beta_over,xi_over,flag_over);
```

执行程序后结果如图 8.13 所示，在 MATLAB 命令窗口中显示下面的信息：

```
max displacement = 0.0206 at eta = 0.5
slope:
  - 0.0618 at eta = 0
  0.0602 at eta = 1
```

① 参见 R.L.Norton, *Design of Machinery*, McGraw-Hill, 1992 年，第 1 003 页。

```
max slope = 0.0618 at eta = 0
max moment = 0.245 at eta = 0.5
shear:
  0.496 at eta = 0
  - 0.514 at eta = 1
max shear = 0.514 at eat = 1
```

该问题用解析方法得到：[①] $y_{max}(\eta=0.5)=0.0208$，$\theta_{max}(\eta=1)=0.0625$，$M_{max}(\eta=0.5)=0.250$，$V_{max}(\eta=0)=0.500$。

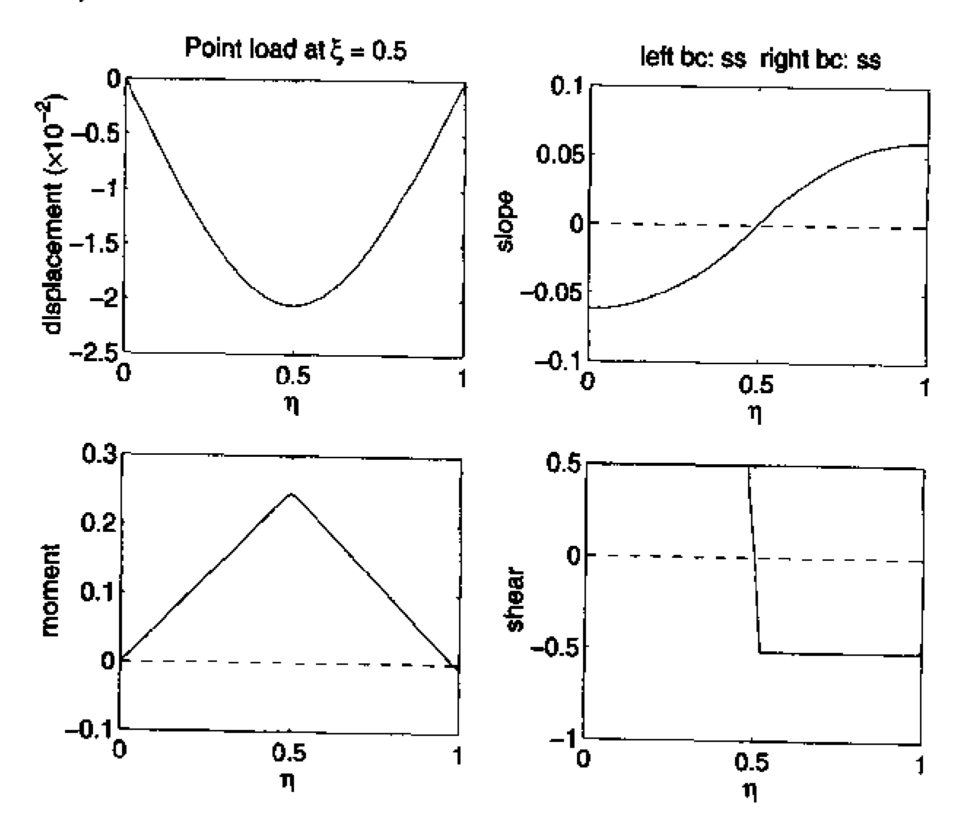


图 8.13　在 $\eta=0.5$ 处有集中载荷作用的简支梁

例 8.13　均布载荷作用的悬臂梁

如果梁在 $\eta=0$ 处固定，$\eta=1$ 处悬空，则这种梁被称为悬臂梁。此梁受 $\eta=0.5$ 至 $\eta=1$ 区间内的均布载荷作用。$\eta=0$ 处的边界条件在表 8.1 的第二行给出，$\eta=1$ 的边界条件在第三行给出。载荷用下列参数描述：$\alpha=1$，$m=0.5$，$n=k=\beta=0=\zeta=0$ 和 $flag=1$，程序清单如下：

```
alpha = 1; m = 0.5; n = 0; k = 0; beta = 0; xi = 0; flag = 1;
alpha_over = 0; m_over = 0; n_over = 0; k_over = 0;
beta_over = 0; xi_over = 0; flag_over = 0;
over = 0; lenover = 0;
options = optimset('display','off');
x = fsolve('BeamSolveCant',[.2.5],options,alpha,m,n,k,beta,xi,flag);
BeamPlot(alpha,m,n,k,beta,xi,flag,[0 0 x(1)x(2)],'clamped','free',over,lenover,...
        alpha_over,m_over,n_over,k_over,beta_over,xi_over,flag_over);
```

① 参见 R.L.Norton，出处同上，第 1003 页。

其中函数 *BeamSolveCant* 是：

```
function d = BeamSolveCant(bc,alpha,m,n,k,beta,xi,flag)
[eta,y] = ode45('StaticBeam',[0:.02:1],[0 0 bc(1) bc(2)]',[],alpha,m,n,k,beta,xi,flag);
d = y[(end,3);y(end,4)];
```

执行程序后结果如图 8.14 所示，在 MATLAB 命令窗口中显示下面的信息：

```
max displacement = 0.108 at eta = 1
slope:
    0 at eta = 0
    - 0.147 at eta = 1
max slope = 0.147 at eta = 1
max moment = 0.379 at eta = 0
shear:
   0.508 at eta = 0
   5.97e - 009 at eta = 1
max shear = 0.508 at eta = 0
```

该问题用解析方法得到①：$y_{max}(\eta=1)=0.106\,8$，$\theta_{max}(\eta=1)=0.145\,8$，$M_{max}(\eta=0)=0.375$，$V_{max}(\eta=0)=0.500$。

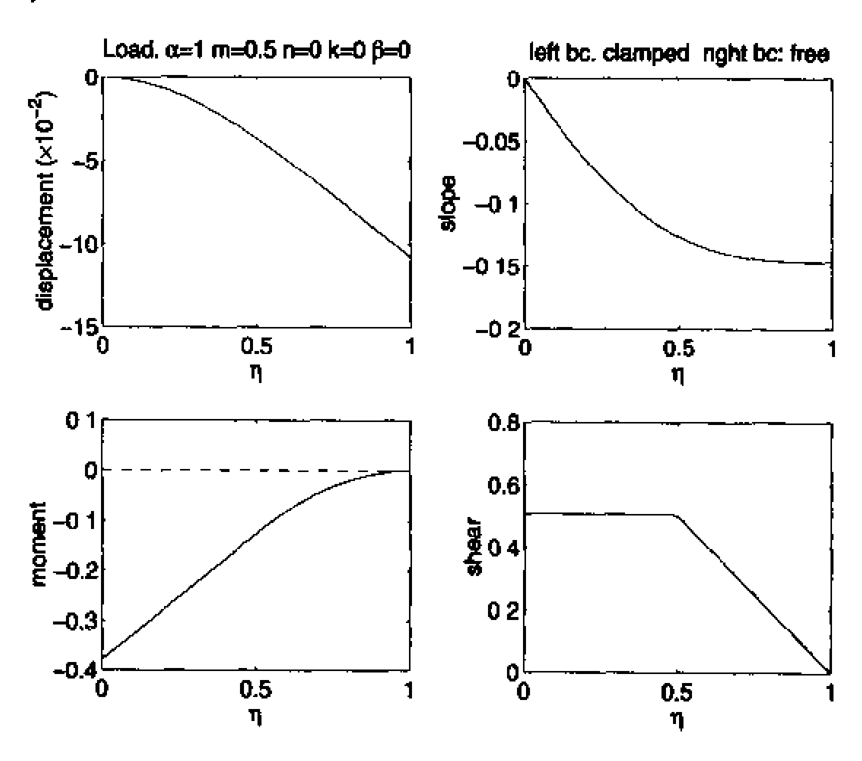


图 8.14 $\eta=0.5$ 至 $\eta=1$ 范围内受均布载荷作用的悬臂梁

例 8.14 集中载荷作用的悬臂梁

梁在 $\eta=0$ 处固定，在 $\eta=1$ 处悬空，集中载荷作用在 $\eta=0.5$ 处。$\eta=0$ 处的边界条件在表 8.1 的第二行给出，$\eta=1$ 的边界条件在第三行给出。载荷用下列参数描述：$\alpha=m=n=k$

① 参见 R.L.Norton，出处同上，第 1 002 页。

$= \beta = 0, \zeta = 0.5$ 和 $flag = 0$,程序清单为:

```
alpha = 0; m = 0; n = 0; k = 0; beta = 0; xi = 0.5; flag = 0;
alpha_over = 0; m_over = 0; n_over = 0; k_over = 0;
beta_over = 0; xi_over = 0; flag_over = 0;
over = 0; lenover = 0;
options = optimset('display','off');
x = fsolve('BeamSolveCant',[.2.5],options,alpha,m,n,k,beta,xi,flag);
BeamPlot(alpha,m,n,k,beta,xi,flag,[0 0 x(1) x(2)],'clamped','free',over,lenover,...
         alpha_over,m_over,n_over,k_over,beta_over,xi_over,flag_over);
```

执行程序后结果如图 8.15 所示,在 MATLAB 命令窗口中显示下面的信息:

```
max displacement = 0.106 at eta = 1
slope:
   0 at eta = 0
   -0.13 at eta = 1
max slope = 0.13 at eta = 1
max moment = 0.496 at eta = 0
shear:
  0.97 at eta = 0
  0.0162 at eta = 1
max shear = 0.97 at eta = 0
```

对这个问题用解析方法[①]得: $y_{max}(\eta = 1) = 0.1042$, $\theta_{max}(\eta = 0.5$ 到 $\eta = 0.5) = 0.1250$, $M_{max}(\eta = 0) = 0.500$ 和 $V_{max}(\eta = 0) = 1.000$。

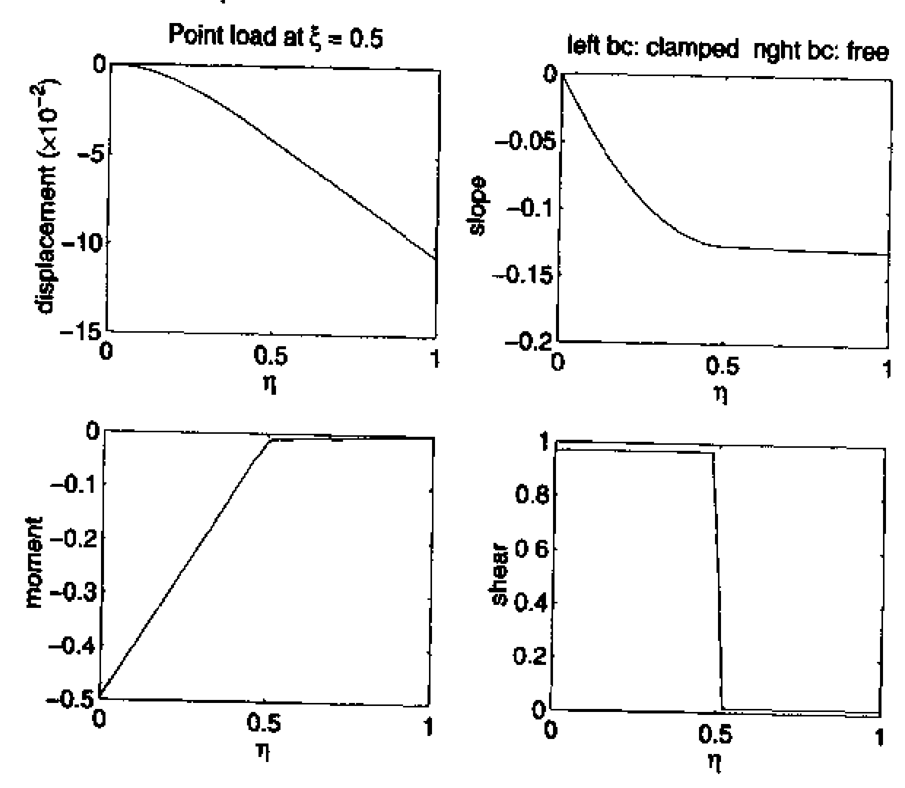


图 8.15　在 $\eta = 0.5$ 处受集中载荷作用的悬臂梁

① 参见 R.L.Norton,出处同上,第 1 002 页。

8.2.2 外伸梁

对第 8.2.1 节中的求值过程加以引申获得如图 8.16 所示的外伸梁的近似数值。首先把一个梁分为两部分：一部分用变量 η 表示，从 0 到 1，无量纲位移记为 $y(\eta)$；另一部分用变量 ζ 表示，从 0 到 ρ，无量纲位移记为 $z(\zeta)$，其中 $\rho>0$。如果假设外伸部分延伸到支撑点($\eta=1$)右侧，则必须满足下面的条件：

1. y 在 $\eta=0$ 处的两个边界条件。
2. z 在 $\zeta=\rho$ 处的自由端边界条件(梁右端的外伸部分)，即：
$$d^2z(\rho)/d\zeta^2 = d^3z(\rho)/d\zeta^3 = 0$$
3. 二梁相接处位移和转角的连续性，即：
$$y(1)=z(0)=0$$
$$dy(1)/d\eta=dz(0)/d\zeta$$
4. 二梁相接处力矩相等，即：
$$d^2z(0)/d\zeta^2=d^2y(1)/d\eta^2$$

反复使用以上条件推出两个方程组。第一个方程组包括 $y(\eta)$，要求在 $\eta=0$ 和 $\eta=1$ 处的边界条件相等，即：

$$y(1)=0$$
$$dy^2(1)/d\eta^2-dz^2(0)/d\zeta^2=0$$

第二个方程组包括 $z(\zeta)$，要求 $d^2z(0)/d\zeta^2$ 和 $d^3z(0)/d\zeta^3$ 在 $\zeta=\rho$ 处相等：

$$d^2z(\rho)/d\zeta^2=d^3z(\rho)/d\zeta^3=0$$

且在 $\zeta=0$ 处为：

$$dy(1)/d\eta=dz(0)/d\zeta$$

下面说明这一过程。

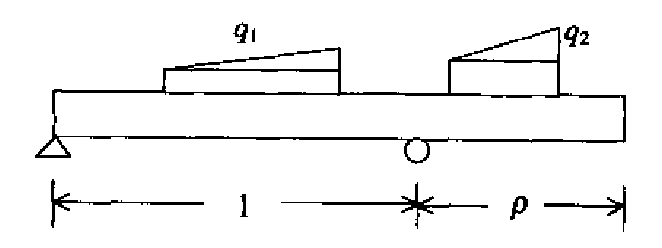


图 8.16 外 伸 梁

例 8.15 均布载荷作用的简支梁

梁在两端简单支撑，$\eta=1$ 处有一个长度为 0.5 的悬臂，即 $\rho=0.5$。梁和悬臂受到沿长度均布载荷的作用。梁的支撑部分载荷用下列参数描述：$\alpha=1, m=n=k=\beta=\zeta=0$ 和 $flag=1$。梁的悬臂部分载荷也用下列参数描述：$\alpha=1, m=n=k=\beta=\zeta=0$ 和 $flag=1$，程序清单如下：

```
alpha = 1; m = 0; n = 0; k = 0; beta = 0; xi = 0; flag = 1;
alpha _ over = 1; m _ over = 0; n _ over = 0; k _ over = 0;
beta _ over = 0; xi _ over = 0; flag _ over = 1;
over = 1; lenover = 0.5;
```

```
options = optimset('display','off');
x = fsolve('BeamSolveOver',[0.5 0.5 0.5 0.5],options,alpha,m,n,k,beta,xi,flag'...
        alpha_over,m_over,n_over,k_over,beta_over,xi_over,flag_over,lenover);
BeamPlot(alpha,m,n,k,beta,xi,flag,[x(1)x(2)x(3)x(4)],'ss','overhung',over,lenover,...
        alpha_over,m_over,n_over,k_over,beta_over,xi_over,flag_over);
```

函数 *BeamSolveOver* 的清单如下：

```
function d = BeamSolveOver(bc,alpha,m,n,k,beta,xi,flag,...
        alpha_over,m_over,n_over,k_over,beta_over,xi_over,flag_over,...
        lenover)
[eta,y] = ode45('StaticBeam',[0:.02:1],[0 bc(1) 0 bc(2)]',[],alpha,m,n,k,beta,xi,flag);
[zeta,z] = ode45 ('StaticBeam',[0:.02:lenover],[0 y(end,2)bc(3) bc(4)],[],...
        alpha_over,m_over,n_over,k_over,beta_over,xi_over,flag_over);
d = [y(end,1);y(end,3) - z(1,3);z(end,3);z(end,4)];
```

执行程序后结果如图 8.17 所示，在 MATLAB 命令窗口中显示下面的信息：

```
max displacement = 0.00781 at eta = 1.5
slope:
  - 0.0208 at eta = 0
  - 0.0208 at eta = 1
max slope = 0.0208 at eta = 1.5
max moment = 0.125 at eta = 1
shear:
0.375 at eta = 0
1.23e - 006 at eta = 1
max shear = 0.625 at eta = 1
```

该问题用解析方法得到：[1] $y_{max}(\eta = 1.5) = 0.007\,8$。

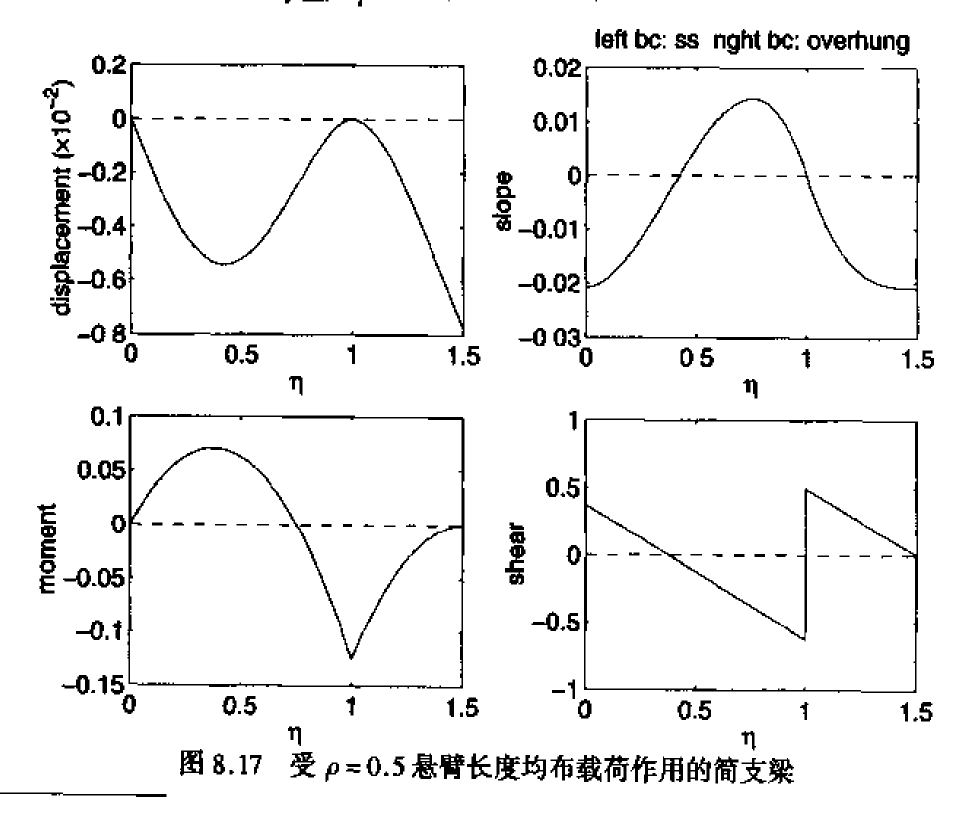


图 8.17　受 $\rho = 0.5$ 悬臂长度均布载荷作用的简支梁

[1] 参见 R. L. Norton，出处同上，第 1 003 页。

8.2.3 柱的屈曲

研究一个长度为 L,横截面积为 A,弹性模量为 E,惯量矩为 I,屈服强度为 S_y 的构造柱,它受到轴向压缩载荷 P,该载荷和柱轴同心并且通过 A 的质心。临界载荷 P 的值记为 P_{cr},用下式估算:

$S_r > \pi\sqrt{2E/S_y}$

$$P_{cr} = \frac{\pi^2 AE}{S_r^2}$$

$S_r \leqslant \pi\sqrt{2E/S_y}$

$$P_{cr} = A\left[S_y - \frac{1}{E}\left(\frac{S_y S_r}{2\pi}\right)^2\right]$$

其中 $S_r = L_{eff}/k$,$k = \sqrt{I/A}$是横截面的惯性半径,对悬臂压杆 $L_{eff} = 2.1L$;对两端铰接柱 $L_{eff} = L$;对一端固定,一端铰接柱 $L_{eff} = 0.8L$;对双端固定柱 $L_{eff} = 0.65L$。

对于一个实心矩形横截面,$k = h/2\sqrt{3}$,其中 h 是横截面的高度;对于一个实心圆形横截面,$k = r/2$,其中 r 是圆的半径;对于一个内径 r_i 和外径 r_o 的圆环,为:

$$k = 0.5\sqrt{r_o^2 + r_i^2}$$

或者,如果 $r_o = r_i + t$,则:

$$k = 0.5\sqrt{r_o^2 + (r_o - t)^2}$$

如果使用安全系数 F_s,则 $P = P_{cr}/F_s$。

当 P 是偏移量为 e 的偏心载荷时,则,

$$P_{cr} = AS_y\left[1 + \frac{ec}{k^2}\sec\left(\frac{l_{eff}}{k}\sqrt{\frac{P_{cr}}{4AE}}\right)\right]^{-1}$$

其中 c 是从截面 A 的形心到截面 A 的外周的距离。当 $ec/k^2 \leqslant 0.025$ 时,该关系有效。当截面是圆周时,$c = r_o$。P_{cr}在等式的两端出现,所以必须使用迭代求解(即使用 fzero)。

下面通过两个例子来说明这些关系。

例 8.16 确定圆柱直径

一个 13 ft 的悬臂钢圆柱受 150 000 lb 的轴向压力载荷作用。如果圆柱的截面是壁厚为 0.75 in 的空心圆。那么当 $E = 3\times10^7$psi,$S_y = 55\,000$ psi 和 $F_s = 3$ 时,确定最小外直径。使用 fzero 来获得 d 的估计值。首先创建一个函数 *ColumnBuckling*:

```
function Pcr = ColumnBuckling(r,P,Sy,E,Leff,Fs,t)
Sr = 2*Leff/sqrt(r^2 + (r - t)^2);
if Sr > pi*sqrt(2*E/Sy)
  Pcr = P - (pi^3*(r^2 - (r - t)^2)*E/Sr^2)/Fs;
else
  Pcr = P - pi*((Sy - ((Sy*Sr)/(2*pi))^2/E) * (r^2 - (r - t)^2))/Fs;
end
```

程序为：

```
options = optimset('display', off');
diam = 2* fzero('ColumnBuckling',5,options,1.5e+5,5.5e+4,3e+7,2.1* 12* 13,3,0.75)
```

程序执行后得到 $d = 8.939\,2$ in，对应于 $P_{cr} = 450\,000$ lb。

例 8.17　确定受偏心载荷作用的圆柱直径

再考虑前一个例子，圆柱偏心载荷的偏移量 $e = 0.6$ in。对于这种情况，要创建两个函数：一个估计给定 r 时的 Pcr，另一个确定 $P = P_{cr}/F_s$ 时的 r。第一个函数是：

```
function Pest = SecColumnBuckling(Pcr,r,t,Sy,E,Leff,ecc)
a = (r^2 - (r - t)^2)* pi;
k = 0.5 * sqrt(r^2 + (r - t)^2);
Pest = Pcr - a* Sy/(1 + ((ecc* r/k)^2)* sec(Leff/k* sqrt(Pcr/4/E/a)));
```

第二个函数是：

```
function Pcr = EccenColumnBuckling(r,p,Sy,E,Leff,Fs,t,ecc)
options = optimset('display',off);
Pcr = p - fzero('SecColumnBuckling',50000,options,r,t,Sy,E,Leff,ecc)/Fs;
```

程序清单如下：

```
options = optimset('display','off');
diam = 2* fzero('EccenColumnBuckling',6,options,15e4,55e3,3e7,...)
                2.1* 12* 13,3,.75,0.6)
```

执行程序后得到 $diam = d = 10.953\,9$ in，对应于 $P_{cr} = 450\,000$ lb。

8.2.4　受交变载荷作用的轴

考虑一个实心圆钢轴分别受到一个完全相反变化的转矩 T_a 和弯曲力矩 M_a 的作用，平均转矩为 T_m 和平均弯曲力矩为 M_m；并且没有轴向载荷，它的直径可以用下式估计：①

$$d = \sqrt[3]{\frac{32F_s}{\pi}\sqrt{\sqrt{(k_fM_a)^2 + 0.75(k_{fs}T_a)^2}/S_f + \sqrt{(k_{fm}M_m)^2 + 0.75(k_{fsm}T_m)^2}/S_{ut}}} \quad (8.15)$$

其中 F_s 是安全系数，S_{ut} 是材料的极限强度，S_f 是修正的疲劳强度，k_a 是变化应力的集中系数。当 M_a/M_m 和 T_a/T_m 为常数时，上式成立。很多值都是 d 的函数，这些函数定义如下：

修正疲劳强度 S_f 为：

$$S_f = C_{size}C_{surf}C_{rel}S_{fu}$$

其中对于圆形轴：

$C_{size} = 1$　　　　$d < 0.3$ in

$C_{size} = 0.869d^{-0.097}$　　　　$0.3 \leqslant d \leqslant 10$ in

$C_{size} = 0.6$　　　　$d > 10$ in

---

① 参见 R.L.Norton，出处同上，第 575 页。

对于机械表面 $C_{surf}=2.7(S_{ut})^{-0.625}$且可靠性为 99% 时，$C_{rel}=0.814$。在缺少相应数据的情况下，采用不精确的疲劳强度 S_{fu}和下面的粗糙度近似值应用于钢材：当 $S_{ut}<200\,000$ psi 时，$S_{fu}=0.5S_{ut}$。基于这些假设，得到：

$$S_f=1.0989C_{size}S_{ut}^{0.735}$$

应力集中系数 k_a 为：

应力集中疲劳系数由下式获得：

$$k_f=1+q(k_t-1) \tag{8.16}$$

其中 k_t 是理论上的静态应力集中系数，q 是槽口灵敏度，是纽博尔常数 a 的函数（参见第 5.6.1 节）：

$$q=\left(1+\sqrt{a/r}\right)^{-1}$$

其中 r 是槽口半径。

k_t 是载荷和几何形状的函数。研究轴直径突然从 D 变为 d 的情况，如图 8.18 所示。弯曲疲劳应力集中系数的值由下式估计：①

$$k_t=1+\left[At_r^{-k}+B\left[\frac{1+a_r}{a_r^{3/2}}\right]^l+\frac{Ca_r}{(a_r+t_r)t_r^m}\right]^{-1/2} \tag{8.17}$$

其中，

$$t_r=t/r_{fillet} \qquad a_r=d/(2r_{fillet})$$

其中 $r_{fillet}<(D-d)/2$。

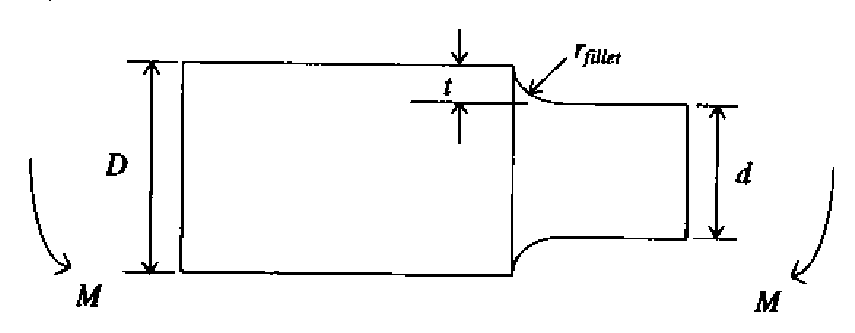


图 8.18 确定应力集中系数的几何尺寸和载荷

在不同弯曲和转矩条件下，A,B,C,k,l,m 的值由表 8.2 给定。

表 8.2 对应理论应力集中系数 k_t 常数的数值

常数	弯曲	转矩
A	0.40	0 40
B	6.00	25.0
C	0 80	0.20
k	0.40	0.45
l	2.75	2.25
m	1.50	2.00

$k_f|\sigma_{max}|<S_y$

$$k_{fm}=k_f$$

① 参见 W. Beitz 和 K. H. Kuttner 编辑，*Handbook of Mechanical Engineering*，Springer-Verlag，New York，1994 年，第 D78 页。

$k_f|\sigma_{max}| > S_y$

$$k_{fm} = (S_y - k_f\sigma_a)/|\sigma_m|$$

$k_f|\sigma_{max} - \sigma_{max}| > 2S_y$

$$k_{fm} = 0$$

其中 k_f 的值由关于弯曲的式(8.16)和式(8.17)确定，上面的方程中给出的应力由下式决定：

$$\sigma_{max} = \frac{M_{max} r_{shaft}}{I} \qquad \sigma_{min} = \frac{M_{min} r_{shaft}}{I}$$

$$\sigma_a = \frac{M_a r_{shaft}}{I} \qquad \sigma_m = \frac{M_m r_{shaft}}{I}$$

其中，$r_{shaft} = d/2$。圆形截面轴的惯性力矩 $I = \pi d^4/64$，M_{max}是最大弯曲力矩，M_{min}是最小弯曲力矩，并且，

$$M_a = (M_{max} - M_{min})/2$$

$$M_m = (M_{max} + M_{min})/2$$

k_{fsm}的值由下式决定：

$k_{fs}|\tau_{max}| < S_s$

$$k_{fsm} = k_{fs}$$

$k_{fs}|\tau_{max}| > S_s$

$$k_{fsm} = (S_s - k_{fs}\tau_a)/|\tau_m|$$

$k_{fs}|\tau_{max} - \tau_{max}| > 2S_s$

$$k_{fsm} = 0$$

其中 k_{fs}的值由转矩式(8.16)和式(8.17)确定，数值 S_s 是剪应力屈服强度，可由下式近似估计：

$$S_s \approx 0.58 S_y$$

上面公式里的应力由下式给出：

$$\tau_{max} = \frac{T_{max} r_{shaft}}{J} \qquad \tau_{min} = \frac{T_{min} r_{shaft}}{J}$$

$$\tau_a = \frac{T_a r_{shaft}}{J} \qquad \tau_m = \frac{T_m r_{shaft}}{J}$$

其中 $J = \pi d^4/32$ 是圆轴截面的惯性矩，T_{max}是最大转矩，T_{min}是最小转矩，并且，

$$T_a = (T_{max} - T_{min})/2$$

$$T_m = (T_{max} + T_{min})/2$$

通过如下的例子可说明这些关系。

例 8.18　受交变载荷作用的轴的直径

一个机械钢圆轴受到的最大弯曲力矩和最小弯曲力矩分别为 4 000 lb in 和 1 000 lb in，并且受到最大和最小转矩分别为 1 600 lb in 和 250 lb in。另外，安全系数是 2.5，材料的屈服强度是 40 000 psi，极限强度为 70 000 psi，槽口半径是 0.03 in。D 比 d 大 15%，圆角半径是 d 的 10%，可靠性为 99%，此时确定轴的最小直径。

因为在式(8.15)中很多数值是轴直径的函数，所以必须根据直径的不同创建几个函数，并

且使用 fzero 来估计 d。这些函数如下：

函数 neuber 计算 q(参见第 5.6.1 节)
函数 StressConcenB 计算 k_f 和 k_{fs}
函数 StressConcenM 计算 k_{fm} 和 k_{fsm}
函数 SsubF，确定正确的疲劳应力 S_f
函数 FatigueDiameter 计算直径 d

钢的 neuber 函数是：

```
function q = neuber(r,Su)
ncs = [50,.13;70,.092;90,.072;110,.057;130,.046;150,.037;...
      170,.028;190,.020;210,.015;230,.010;250,.007];
q = 1/(1 + polyval(polyfit(ncs(:,1),ncs(:,2),4),Su)/sqrt(r));
```

其中 ncs 的值在表 5.1 中给出。
函数 StressConcenB 是：

```
function [kf,kfs] = StressConcenB(d,rfillet,dout,rnotch,Su)
A = [0.4 0.4];B = [6 25];C = [0.8 0.2];
k = [0.4 0.45];l = [2.75 2.25];m = [1.5 2];
dod = dout/d;
rd = rfillet/d;
tr = (dout - d)/2/rfillet;
ar = d/rfillet/2;
for n = 1:2
  t1 = A(n)/(tr)^k(n);
  t2 = B(n)* *((1 + ar)/(ar* sqrt(ar)))^l(n);
  t3 = C(n)* ar/(ar + tr)/(tr)^m(n);
  alpha(n) = 1 + 1/sqrt(t1 + t2 + t3);
end
kf = 1 + neuber(rnotch,Su)* (alpha(1) - 1);
kfs = 1 + neuber(rnotch,Su)* (alpha(2) - 1);
```

函数 StressConcenM 是：

```
function[kfm,kfsm] = StressConcenM(d,Mmax,Mmin,Tmax,Tmin,Sy,kf,kfs)
roj = 16/pi/d^3;
roi = 2* roj;
Smax = Mmax* roi;
Smin = Mmin* roi;
if kf* abs(Smax) <= Sy
 kfm = kf;
else
 kfm = (Sy - kf* roi* (Mmax - Mmin)/2)/abs(roi* (Mmax + Mmin)/2);
end
if kf* abs(Smax - Smin) > 2* Sy;kfm = 0;end
Tmax = Tmax* roj;
Tmin = Tmin* roj;
if kf* abs(Tmax) <= 0.58* Sy
 kfsm = kfs;
```

```
else
  kfsm = (.58* Sy - kfs* roj* (Tmax - Tmin)/2)/abs(roj* (Tmax + Tmin)/2);
end
if kf* abs(Tmax - Tmin) > 1.16* Sy; kfsm = 0; end
```

函数 SsubF 是：

```
function Sf = SsubF(d,Su)
csize = 0.869* d^(-0.097);
if d < 3; csize = 1; end
if d > 10; csize = 0.6; end
Sf = 1.0989* csize* Su^(0.735);
```

函数 FatigueDiameter 是：

```
function diam = FatigueDiameter(d,Mmax,Mmin,Tmax,Tmin,Sy,Su,...
                prfillet,pDod,rnotch,fs)
Sf = SsubF(d,Su);
dout = pDod* d;
rfill = prfillet* d;
[kf,kfs] = StressConcenB(d,rfill,dout,rnotch,Su);
[kfm,kfsm] = StressConcenM(d,Mmax,Mmin,Tmax,Tmin;Sy,kf,kfs);
p1 = sqrt((kf* (Mmax - Mmin)/2)^2 + 0.75* (kfs* (Tmax - Tmin)/2)^2)/Sf;
p2 = sqrt((kfm* (Mmax + Mmin)/2)^2 + 0.75* (kfsm* (Tmax + Tmin)/2)^2)/Su;
diam = d - (32/pi* fs* (p1 + p2))^(1/3);
```

其中 $prfillet = r_{fillet}/d$，$pDod = D/d$。

程序清单是：

```
options = optimset('display','off');
d = fzero('FatigueDiameter',3,options,4e3,1e3,1.6e3,250,4e4,7e4,0.1,1 15,0.03,2 5)
```

执行程序得到：$d = 2.2376$ in。

8.3　直齿圆柱齿轮的应力

在一个齿轮齿上的弯曲应力取决于一个均布(不变)的切向传输载荷 F_t，该弯曲应力由下式给出：

$$\sigma_b = \frac{K_v K_H F_t}{mbJ_K} \text{N/mm}^2$$

其中 b 是齿轮齿的表面宽度，m 是模量，即 J_k 是弯曲强度的几何系数。K_v 是动载系数，该动载系数与轮齿的质量和齿节的切向速度 v_t 有关，K_H 是载荷分布系数。

切向载荷可由下式求出：

$$F_t = \frac{9.549 \times 10^6 P}{nR_p} = \frac{1\,000\,T}{R_p} \text{ N}$$

其中 P 是功率，单位是 kW；T 是扭矩，单位是 Nm；n 是两个齿轮中较小的一个齿轮转速，单位

是 r/m。R_p 是两个齿轮中较小的操作齿节距，单位是 mm。

动载系数由下式估计：

$$K_v = \left(\frac{A + \sqrt{200 v_t}}{A}\right)^B$$

其中，

$$v_t = \frac{2\pi R_p n}{60\,000} \quad \text{m/s}$$

$$A = 50 + 56(1 - B)$$

$$B = 0.25(12 - Q_v)^{2/3}$$

Q_v 是齿轮的质量系数，取 $5 \leqslant Q_v \leqslant 11$ 区间内的整数。上限用于精密齿轮，下限用于最低精密度的齿轮。工作节圆半径是 R_p，单位为 mm；n 是转速，单位 r/m；对于每一个 Q_v 值，V_t 有一个建议值，由下式给出：

$$v_{t_{\max}} = (A + Q_v - 3)^2/200 \text{ m/s}$$

设计安装在轴承间刚性齿轮时，不考虑外部相对偏差的影响。载荷分布系数可由下式估算：

$$K_H = 1 + K_{Hpf} + K_{Hma}$$

其中 K_{Hpf} 是一个小齿轮的比例系数，K_{Hma} 是啮合补偿系数。小齿轮的比例系数由下式估算：

$b \leqslant 25$ mm：

$$K_{Hpf} = k_o - 0.025$$

$25 < b \leqslant 432$ mm：

$$K_{Hpf} = k_0 - 0.0375 + 0.000\,492 b$$

$432 < b \leqslant 1020$ mm：

$$K_{Hpf} = k_o - 0.1109 + 0.000\,815 b - 0.353 \times 10^{-6} b^2$$

其中 $k_o \geqslant 0.05$ 时，

$$k_o = 0.05 b / R_p$$

其中 $k_o < 0.05$ 时，

$$k_o = 0.05$$

数值 R_p 是两个轮中小轮的工作节圆半径。

啮合补偿系数由下式估算：

$$K_{Hma} = A + Bb + Cb^2$$

其中经验数据 A、B 和 C 由表 8.3 给出。

表 8.3 常数 A、B 和 C*

分类号	齿轮类型	A	B	C
1	开式	0.247	0.657×10^{-3}	-1.186×10^{-7}
2	闭式	0.127	0.622×10^{-3}	-1.69×10^{-7}
3	精密闭式	0.0675	0.504×10^{-3}	-1.44×10^{-7}
4	超精密闭式	0.0380	0.402×10^{-3}	-1.27×10^{-7}

* 引自 AGMA 标准 2101-C95。

确定几何系数是很繁琐的，然而，AGMA 提供了一种图形化方法。将要用它来代替由 Colbourne[①] 提出的计算方法。AGMA 的分析过程用于确定直齿圆柱齿轮的几何系数，不同的仅是找出轮齿临界的截面圆角。

计算过程中需要大量的术语和表达式，在表 8.4 和表 8.5 中列出，并在图 8.19 和图 8.20 中加以说明。参考这些图表，几何系数能用下式估计：

$$J_K = \frac{\cos\varphi}{mK_J\cos\gamma_w}$$

其中，

$$K_J = K_f\left[\frac{1.5(x_D - x)}{y^2} - \frac{\tan\gamma_w}{2y}\right]_{max}$$

$$K_f = k_1 + \left(\frac{2y}{r_f}\right)^{k_2}\left(\frac{2y}{x_D - x}\right)^{k3}$$

并且，

$$k_1 = 0.305\,4 - 0.004\,89\varphi_s - 0.000\,069\varphi_s^2$$

$$k_2 = 0.362\,0 - 0.012\,68\varphi_s + 0.000\,104\varphi_s^2$$

$$k_3 = 0.293\,4 + 0.006\,09\varphi_s + 0.000\,087\varphi_s^2$$

其中 φ_s 是下面的齿轮压力角的度数表达：可以是 14.5°、20°或 25°。中括号$[\cdots]_{max}$内的量表示 K_J 是最大值，由变量 x，y 确定。改变下式中的 u_r 可得到 x 和 y 的值，表 8.5 给出 u_r 的范围，如下：

$$u_{r_{min}} \leqslant u_r \leqslant u_{r_{max}}$$

x 和 y 值由下式给出：

$$x = R\cos\theta_R$$

$$y = R\sin\theta_R$$

其中，

$$R = \sqrt{(N_1 m/2 + \zeta)^2 + \eta^2}$$

$$\theta_R = \arctan\left(\frac{\eta}{N_1 m/2 + \zeta}\right) - \frac{u_r - m\pi/2}{N_1 m/2}$$

$$\zeta = s\zeta' \qquad \eta = s\eta'$$

$$s = 1 + \frac{r_{rT}}{\sqrt{\zeta'^2 + \eta'^2}}$$

$$\zeta' = e + x'_r \quad \eta' = u_r + y'_r$$

① 参见 J.R.Colbourne, *The Geometry of Involute Gears*, Springer - Verlag, New York, 1987 年。

表 8.4 用于确定直齿圆柱齿轮几何系数 J_k 参数的定义

参量	符号/公式
模数	m
齿轮齿数	N
基本节距	$p_b = m\pi\cos(\varphi_s)$
圆弧节距	$p_s = m\pi$
直径节距	$p_d = 1/m$
标准节圆半径	$R_s = Np_s/2\pi = Nm/2$
基圆半径	$R_b = R_s\cos(\varphi_s) = Np_b/2\pi = (Nm/2)\cos(\varphi_s)$
齿顶圆半径(齿坯半径)	R_T
齿顶高	$a = R_T - R_s$
(完全深度齿的标准齿顶高)	$(a = m)$
齿轮廓线上的点半径	R
工作节圆半径	$R_{p1} = C/(1 + N_2/N_1), R_{p2} = C/(1 + N_1/N_2)$
齿根半径	R_{root}
中心距	$C = R_{p1} + R_{p2}$
齿轮压力角	φ_s
齿条压力角	$\varphi_r (= \varphi_s)$
工作压力角	φ
在 R_s 处的齿厚	t_s
在 R 处的齿厚(见图 8.19)	$t_R = 2R\theta_R$
齿根圆角半径	r_f
齿条齿顶高*	a_r
齿条铣刀的顶圆半径*	r_{rT}
参考线和切割节线的距离[参考线在齿厚与齿间宽度相等的位置上(= $m\pi/2$),参见图 8.20。]	e
渐开线函数	$\mathrm{inv}(\varphi) = \tan(\varphi) - \varphi$

* 选择 a_r 和 r_{rT} 的取值,使之近似于与 m 相等,h 在表 8.5 中定义。

表 8.5 计算 J_k 时使用的几何值⁺

$e = \dfrac{t_s - m\pi/2}{2\tan\varphi_s}$	$m_c = \left[\sqrt{R_{T_1}^2 - R_{b_1}^2} - \sqrt{R_{T_2}^2 - R_{b_2}^2} - (R_{b_1} + R_{b_2})\tan\varphi\right] / p_b$
$r_f = r_{rT} + \dfrac{(a_r - e - r_{rT})^2}{mN_1/2 + a_r - e - r_{rT}}$	$R_w^2 = R_b^2 + \left[\sqrt{R_T^2 - R_b^2} - (m_c - 1)p_b\right]^2$
$Rb_j = N_j\pi\cos\varphi_s \quad j = 1,2$	$\varphi_w = \arccos(R_b/R_w)$
$\varphi = \arccos\left(\dfrac{R_{b_1} + R_{b_2}}{C}\right)$	$\theta_w = \dfrac{t_s}{Nm} + \mathrm{inv}\varphi_s - \mathrm{inv}\varphi_w$
$x'_r = -a_r + r_{rT}$	$\gamma_w = \varphi_w - \theta_w$
$y'_r = m\pi/4 + h\tan\varphi_s + r_{rT}\cos\varphi_s$	$x_D = R_w\cos\theta_w - R_w\sin\theta_w\tan\gamma_w$
$h = a_r - r_{rT} + r_{rT}\sin\varphi_s$	$u_{r_{min}} = \dfrac{e + x'_r}{\tan\varphi_s} - y'_r$
	$u_{r_{max}} = -y'_r$

⁺ 下标"1"表示要确定 J_k 的齿轮,下标"2"表示啮合齿轮。不带下标表示正计算 J_k 的齿轮。

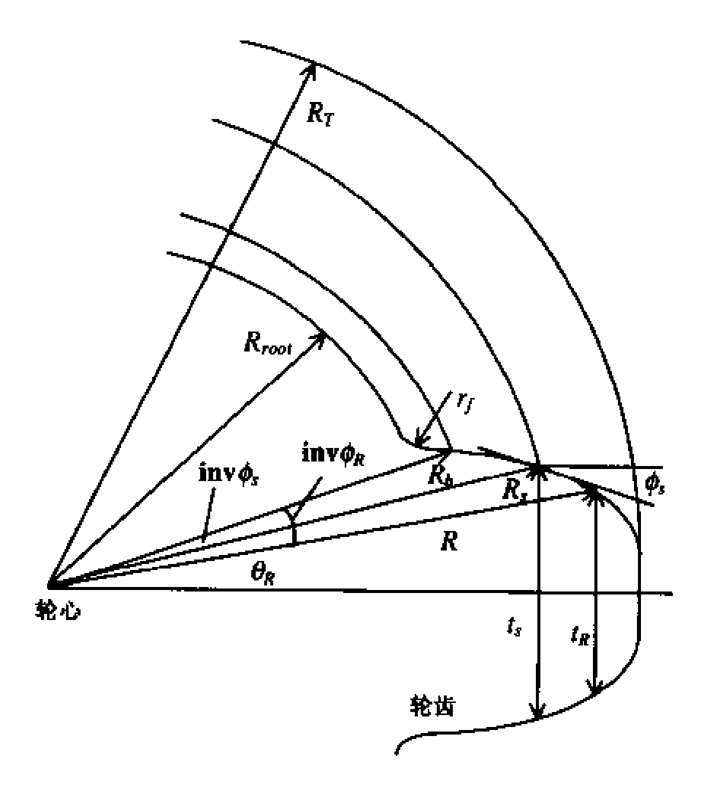


图 8.19　关于齿轮的术语

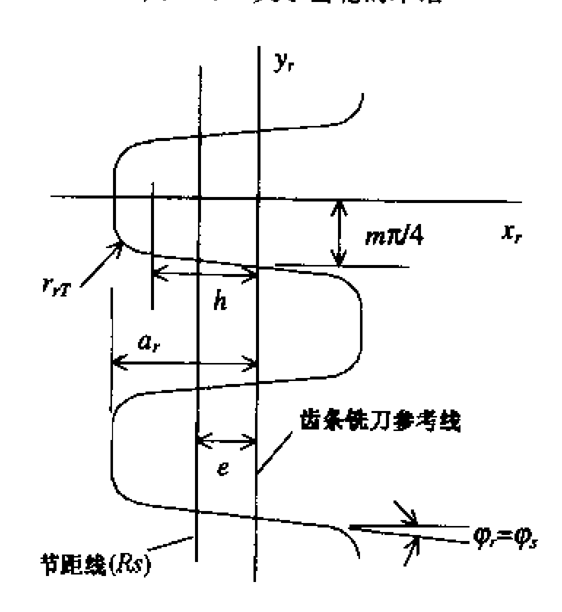


图 8.20　关于轮齿齿条铣刀的术语

一旦应力 σ_b 被计算出来，许用应力 σ_F 便被确定，并验证是否比由 F_t 引起的应力大，即 $\sigma_b \leqslant \sigma_F$。其中许用应力是：

$$\sigma_F = \frac{\sigma_{FP} Y_N}{F_S Y_Z}$$

上式适用于油或齿轮温度低于 120℃的情况，数值 F_s 是安全系数，Y_N 是屈服应力循环系数，Y_Z 是可靠系数。可靠系数有如下值：$Y_Z = 1$ 表示失效率小于 1%，$Y_Z = 1.25$ 表示失效率小于 0.1%，$Y_Z = 1.5$ 表示失效率小于 0.01%。

如果 n_L 是单向齿载荷循环次数，那么在 99% 的可靠性内，钢齿轮的弯曲应力循环载荷系数可由下式估计：

$n_L \leqslant 3\times 10^3$:

$$Y_N = f(B_H)$$

其中,

$$f(B_H) = -9.2592\times 10^{-6}(B_H-160)^2 + 0.009\,722(B_H-160) + 1.6 \qquad 160 \leqslant B_H \leqslant 400$$

而 B_H 是布里涅耳(Brinell)硬度值。

$3\times 10^3 \leqslant n_L \leqslant 3\times 10^6$:

$$Y_N = Dn_L^E$$

其中,

$$D = f(B_H)10^{0.862\,8C_1}$$
$$E = -0.287\,6C_1$$
$$C_1 = \log_{10}[f(B_H)] - 0.016\,9$$

$n_L > 3\times 10^6$:

$$Y_N = 1.638n_L^{-0.032\,3}$$

式中 B_H 是独立的。

循环次数由下式获得:

$$n_L = 60nL$$

其中 L 是齿轮的寿命,单位按小时计;n 是齿轮的转速,单位是 r/m;并且假设节数是一。

对一级硬钢齿轮的许用弯曲应力数值是:

$$\sigma_{FP} = 0.533B_H + 88.3\ \text{N/mm}^2$$

在举例之前,有必要创建如下的函数来计算不同的系数:

函数 *GearKofV* 计算 K_v

函数 *GearKofH* 计算 K_H

函数 *involute* 计算角的渐开

函数 *GearParameters* 计算表 8.5 中的数值

函数 *GearKofF* 计算 K_f

函数 *GearKofJ* 计算 K_J

函数 *GearJofK* 计算 J_K

函数 *GearYofK* 计算 Y_N

函数 *GearKofV* 是:

```
function Kv = GearKofV(Rp,n,Qv)
vt = 2 * pi * Rp * n/60000;
B = 0.25 * (12 - Qv)^(2/3);
A = 50 + 56 * (1 - B);
vtmax = (A + Qv - 3)^2/200;
if vt > vtmax
 error('Maximum tangential velocity exceeded for given Qv')
end
Kv = ((A + sqrt(200 * vt))/A)^B;
```

函数 *GearKofH* 是:

```
function Kh = GearKofH(b,typeg,Rp)
class = [0.247 0.127 0 0675 0.0380;...
        0.657e-3 0.622e-3 0.504e-3 0.402e-3;...
       -1.186e-7 -1.69e-7 -1.44e-7 -1.27e-7];
Khma = class(1,typeg) + class(2,typeg) * b + class(3,typeg) * b^2;
ko = 0.05 * b/Rp;
if ko < 0.05,ko = 0.05;end
if b <= 25
 Khpf = ko - 0.025;
 elseif b <= 432
  Khpf = ko - 0.0375 + 0.000492 * b;
 else
  Khpf = ko - 0.1109 + 0.000815 * b - 0.353e-6 * b^2;
end
Kh = 1 + Khpf + Khma;
```

函数 *involute* 是：

```
function inv = involute(angle)
inv = tan(angle) - angle;
```

函数 *GearParameter* 是：

```
function [urmin,urmax,e,x D,rf,xrp,yrp,gammaw,phi] =
          GearParameters(m,phis,ar,rrT,ts,N1,N2,C,Rt1,Rt2)
xrp = - ar + rrT;
h = ar - rrT + rrT * sin(phis);
yrp = - pi * m/4 + h * tan(phis) + rrT * cos(phis);
e = (ts - pi * m/2)/tan(phis)/2;
urmin = (e + xrp)/tan(phis) - yrp;
urmax = - yrp;
Pb = m * pi * cos(phis);
Rb1 = N1 * m/2 * cos(phis);
Rb2 = N2 * m/2 * cos(phis);
phi = acos((Rb1 + Rb2)/C);
mc = (sqrt(Rt1^2 - Rb1^2) + sqrt(Rt2^2 - Rb2^2) * - (Rb1 + Rb2)* tan(phi))/Pb;
Rw = sqrt(Rb1^2 + (sqrt(Rt1^2 - Rb1^2) - (mc - 1) * Pb)^2);
phiw = acos(Rb1/Rw);
thetaw = ts/m/N1 + involute(phis) - involute(phiw);
gammaw = phiw - thetaw;
xD = RW * cos(thetaw) - Rw * sin(thetaw) * tan(gammaw);
rf = rrT + (ar - e - rrT)^2/(N1 * m/2 + ar - e - rrT);
```

函数 *GearKofF* 是：

```
function Kf = GearKofF(phis,rf,xD,x,y)
d = phis * 180/pi;
k1 = 0.3054 - 0.00489 * d - 0.000069 * d^2;
k2 = 0.362 - 0.01268 * d + 0.000104 * d^2;
k3 = 0.2934 + 0.00609 * d + 0.000087 * d^2;
Kf = k1 + (2 * y/rf)^k2 * (2 * y/(xD - x))^k3;
```

函数 *GearKofJ* 是：

```
function KJ = GearKofJ(ur,m,rf,e,rrT,N1,xrp,yrp,phis,xD,gammaw)
xip = e + xrp;
etap = ur + yrp;
s = 1 + rrT/sqrt(xip^2 + etap^2);
xi = s * xip;
eta = s * etap;
thetaR = atan(eta/(N1 * m/2 + xi)) - (ur - pi * m/2)/(N1 * m/2);
R = sqrt((N1 * m/2 + xi)^2 + eta^2);
x = R * cos(thetaR);
y = R * sin(thetaR);
Kf = GearKofF(phis,rf,xD,x,y);
KJ = - m * cos(gammaw) * Kf * (1.5 * (xD - x)/y^2 - 0.5 * tan(gammaw)/y);
```

在最后一行使用负号的原因参见有关函数 *GearJofK* 的讨论。

函数 *GearJofK* 是：

```
function JK = GearJofK(m,phis,ar,rrT,ts,N1,N2,C,rT1,rT2)
[urmin,urmax,e,xD,rf,xrp,yrp,gammaw,phi] = ...
                    GearParameters(m,phis,ar,rrT,ts,N1,N2,C,rT1,rT2);
options = optimset('display','off');
ur = fminbnd('GearKofJ',urmin,urmax,options,m,rf,e,rrT,N1,xrp,yrp,...
                  phis,xD,gammaw);
JK = - cos(phi)/GearKofJ(ur,m,rf,e,rrT,N1,xrp,yrp,phis,xD,gammaw);
```

该函数使用 fminbnd 确定 K_J 的最大值，K_J 为 u_r 的函数。为了让 fminbnd 正确执行，必须改变函数中 *GearKofJ* 的符号。因此，通过在函数 *GearJofK* 的最后一行插入负号来改变 J_k 的符号。

函数 *GearYofN* 是：

```
function YN = GearYofN(BH,n,L)
nL = 60 * n * L;
fBH = -9.2592e-6 * (BH - 160)^2 + 0.009722 * (BH - 160) + 1.6;
if nL <= 1e3
  YN = fBH;
elseif nL <= 3e6
  D = 0.8628 * (log10(fBH) - 0.0169);
  E = -0.2876 * (log10(fBH) - 0.0169);
  YN = (fBH * 10^D) * nL.^E;
else
  YN = 1.683 * nL^(-0.0323);
end
```

下面在例子中使用这些函数。

例 8.19 轮齿的弯曲强度

考虑如下几对齿轮以及制作这些齿轮的齿条铣刀的几何参数。

$m = 10$ mm	$a_r = 12.5$ mm	$R_{T_1} = 153.9$ mm
$\varphi_s = 20°$	$r_{rT} = 3.8$ mm	$R_{T_2} = 391.1$ mm
$N_1 = 28$	$b = 45$ mm	$n_1 = 1\,800$ r/m

$N_2 = 75$　　$C = 525$ mm　　$B_H = 260$

$T = 2\,500$ Nm　　$t_s = 18.51$ mm

注意到 $h = 10 = m$,在允许的范围之内。假设安全系数是 1.2,齿轮将使用 4 000 小时,期望的失效性低于 1% ,这意味着 $Y_Z = 1.0$。而且,假设 $Q_v = 8$,并将齿轮安装在表 8.3 中类型 2 指定的密闭装置内,确定弯曲应力和许用应力的程序清单如下:

```
m = 10; phis = 20 * pi/180; ar = 12.5; rrT = 3.8; ts = 18.51;
C = 525; N1 = 28; N2 = 75; rT1 = 153.9; rT2 = 391.1; torque = 2400;
BH = 260; fs = 1.2; YZ = 1.0; b. = 45; n = 1800; Qv = 8; typeg = 2; L = 4000;
Rp = C/(1 + N2/N1);
Ft = 1000 * torque/Rp;
JK = GearJofK(m, phis, ar, rrT, ts, N1, N2, C, rT1, rT2);
Kv = GearKofV(Rp, n, Qv);
KH = GearKofH(b, typeg, Rp);
sigmab = Kv * KH * Ft/m/b/JK;
sigmafp = 0.533 * BH + 88.3;
YN = GearYofN(BH, n, L);
sigmaf = sigmafp * YN/fs/YZ;
disp(['The bending stress is' num2str(sigmab,5)'N/mm^2'])
disp(['The permissible level is 'num2str(sigmaf,5)'N/mm^2']
```

程序执行后,在 MATLAB 命令窗口显示如下内容:

```
The bending stress is 149.64 N/mm^2
The permissible level is 167.41 N/mm^2
```

8.4 四连杆机构的运动学

8.4.1 连杆的位置和速度

研究如图 8.21 所示的机构。当角度 θ_1 和 θ_2 给定时,θ_3 和 θ_4 由下式给出:

$$\begin{aligned} L_2\cos\theta_2 + L_3\cos\theta_3 - L_4\cos\theta_4 - L_1\cos\theta_1 &= 0 \\ L_2\sin\theta_2 + L_3\sin\theta_3 - L_4\sin\theta_4 - L_1\sin\theta_1 &= 0 \end{aligned} \tag{8.18}$$

为方便起见,令 $\theta_1 = 0$。如果连杆 2 的角速度给定,则连杆 3 和连杆 4 的角速度分别是:

$$\omega_3 = \frac{L_2\omega_2\sin(\theta_4 - \theta_2)}{L_3\sin(\theta_3 - \theta_4)} \qquad \omega_4 = \frac{L_2\omega_2\sin(\theta_2 - \theta_3)}{L_4\sin(\theta_4 - \theta_3)}$$

其中 θ_3,θ_4 由公式(8.18)得到。连杆 2 端部的线速度是 $V_2 = L_2\omega_2$,连杆 4 端部的线速度是:

$$V_4 = L_4\omega_4 = V_2\frac{\sin(\theta_2 - \theta_3)}{\sin(\theta_4 - \theta_3)}$$

每一种速度在角速度 ω_j 的方向上分别垂直于杆 L_j。

连杆 3 和连杆 4 的角加速度是:[①]

---

① 参见 A.G.Erdman 和 G.N.Sandor, *Mechanism Design: Analysis and Synthesis*,第二版,Prentice Hall,Upper Saddle River,NJ,1991 年,第 231 页.

$$\alpha_3 = \frac{-L_2\alpha_2\sin(\theta_4-\theta_2)+L_2\omega_2^2\cos(\theta_4-\theta_2)+L_3\omega_3^2\cos(\theta_4-\theta_3)-L_4\omega_4^2}{L_3\sin(\theta_3-\theta_4)}$$

$$\alpha_4 = \frac{L_2\alpha_2\sin(\theta_3-\theta_2)-L_2\omega_2^2\cos(\theta_3-\theta_2)+L_4\omega_4^2\cos(\theta_3-\theta_4)-L_3\omega_3^2}{L_4\sin(\theta_3-\theta_4)}$$

用代数求解法可以从式(8.18)直接获得 θ_3 和 θ_4 的表达式，但很烦琐。应该使用 fsolve 函数来求解。

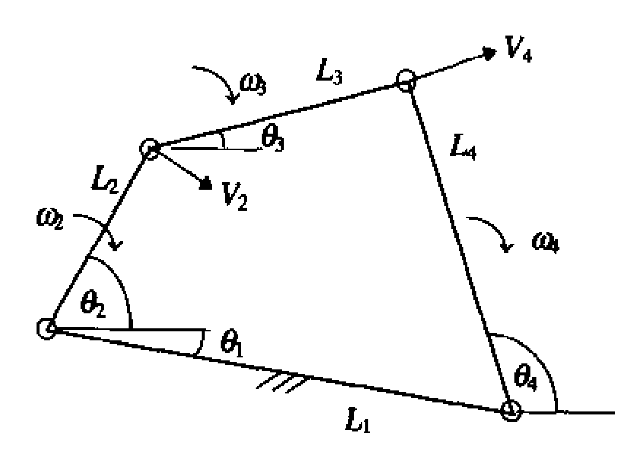


图 8.21 四连杆机构的术语

例 8.20 四连杆机构的位置、速度和加速度的图形化表示

在 $L_1=0.8, L_2=2, L_3=2, L_4=3$ 的情况下，画出连杆 3 的方向，在另一幅图绘出速度比 V_4/V_2 和 $\omega_2=4$ rad/s 及 $\alpha_2=5$ rad/s^2 时的加速度 α_4。首先创建函数 FourBarPosition，函数 fsolve 通过它确定 θ_3 和 θ_4。这样，

```
function t = FourBarPosition(th,th2,L2,L3,L4,L1)
t = [L2 * cos(th2) + L3 * cos(th(1)) - L4 * cos(th(2)) - L1;...
  L2 * sin(th2) + L3 * sin(th(1)) - L4 * sin(th(2))];
```

其中 $th(1)=\theta_3, th(2)=\theta_4$。

显示连杆 3 的位置、连杆 4 的端部速度和加速度的程序如下：

```
L2 = .8;L3 = 2;L4 = 2;L1 = 3;th1 = 0;
th2 = [1/6:1/6:2] * pi;
th34 = zeros(length(th2),2);
options = optimset('display','off');
for m = 1:length(th2)
 th34(m,:) = fsolve('FourBarPosition',[55],options,th2(m),L2,L3,L4,L1);
end
y = L2 * sin(th2) + L3 * sin(th34(:,1)');
x = L2 * cos(th2) + L3 * cos(th34(:,1)');
xx = [L2 * cos(th2)];
yy = [L2 * sin(th2)];
figure(1)
plot([x;xx],[y;yy],'k',[0 L1],[0 0],'k--^',x,y,'ko',xx,yy,'ks')
title('Several positions of the connecting link of a four-bar mechanism')
xlabel('Horizontal position')
ylabel('Vertical position')
axis equal
```

```
th2 = [0:05:2] * pi;
th34 = zeros(length(th2),2);
for m = 1:length(th2)
  th34(m,:) = fsolve('Four BarPosition',[5 5],options,th2(m),L2,L3,L4,L1);
end
figure(2)
y = sin(th2 - th34(:,1)')./sin(th34(:,2)' - th34(:,1)');
plot(180 * th2/pi,y)
v = axis;
v(2) = 360;
axis(v)
xlabel('\theta_2(degrees)')
ylabel('V_4/V_2')
title('Velocity ratio of tip of link #4')
w2 = 4;alph2 = 5;
w3 = (L2 * w2 * sin(th34(:,2) - th2'))./(L3 * sin(th34(:,1) - th34(:,2)));
w4 = (L2 * w2 * sin(th2' - th34(:,1)))./(L4 * sin(th34(:,2) - th34(:,1)));
s32 = th34(:,1) - th2';
s34 = th34(:,1) - th34(:,2);
alph4 = (L2 * alph2 * sin(s32) - L2 * w2^2 * cos(s32) + L4 * w4.^2. * cos(s34) - ...
      L3 * w3.^2)./(L4 * sin(s34));
figure(3)
plot(180 * th2/pi,alph4)
v = axis;
v(2) = 360;
axis(v)
xlabel('\theta_2(degrees)')
ylabel('\alpha_4')
title('Angular acceleration of link #4')
```

程序执行结果如图 8.22、图 8.23 和图 8.24 所示。

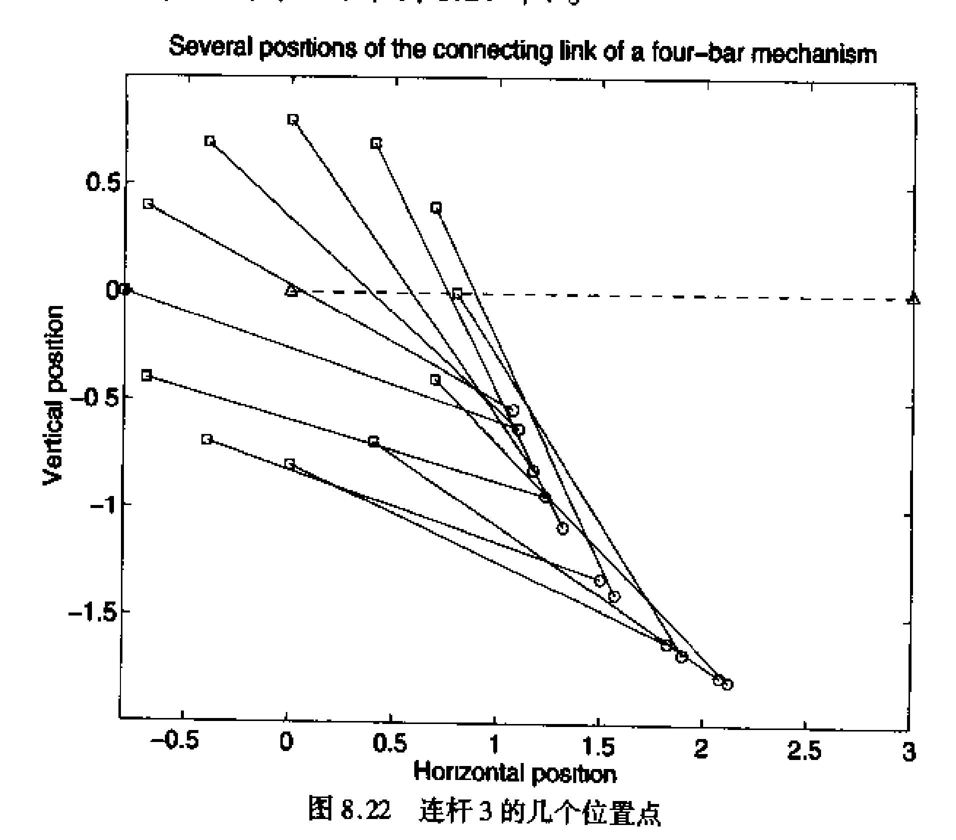


图 8.22　连杆 3 的几个位置点

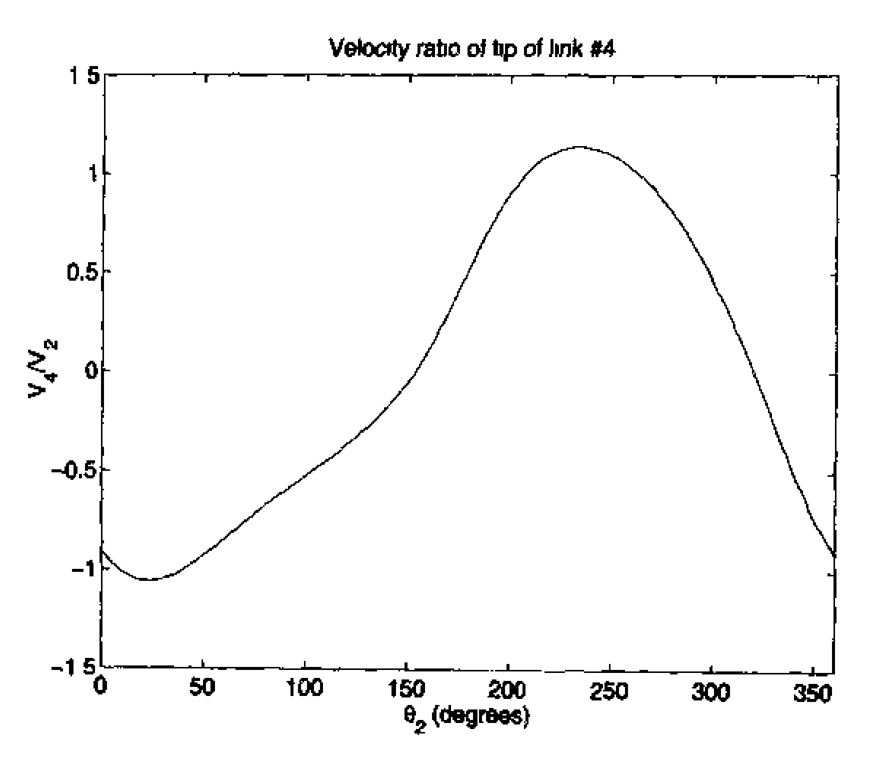


图 8.23　连杆 4 的端部速度

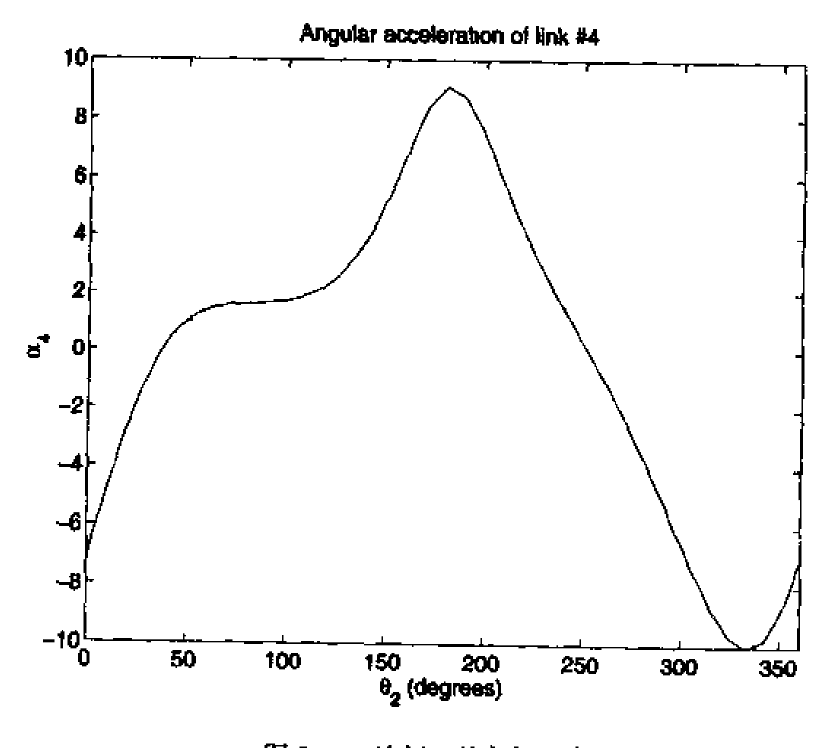


图 8.24　连杆 4 的角加速度

8.4.2 四连杆机构的综合

四连杆机构综合的目标之一是确定连杆的长度，使游动杆上选定的点通过三个规定的点。研究如图 8.25 所示的机构。目标是确定长度 Z_k 及其初始方向，以使点 P 通过点 P_1、P_2 和 P_3。假设点 P_1 在坐标系的原点处，使 $R_1=0$。

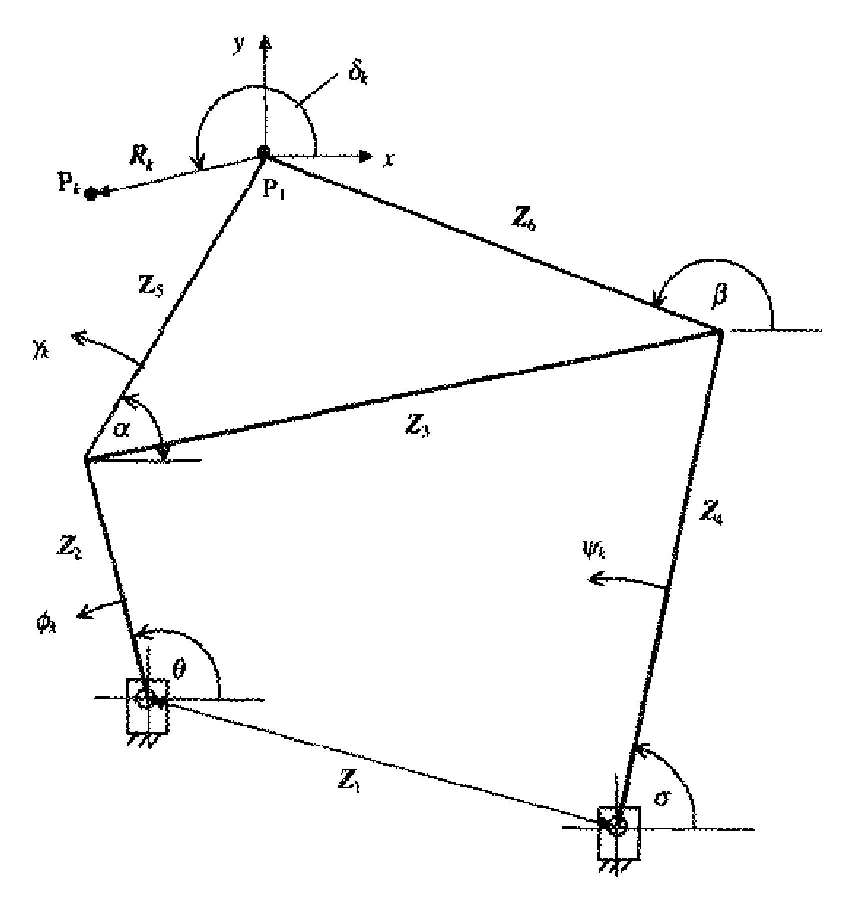


图 8 25 三点综合的向量定义

计算六个连杆长度的方程用二维向量给出，该向量可以很容易地表示为复数。六个向量的长度和方向可以由下式确定：①

$$\boldsymbol{Z}_2 = \frac{1}{\boldsymbol{D}}\begin{vmatrix} \boldsymbol{R}_1 & \boldsymbol{d}_{12} \\ \boldsymbol{R}_2 & \boldsymbol{d}_{22} \end{vmatrix} \qquad \boldsymbol{Z}_5 = \frac{1}{\boldsymbol{D}}\begin{vmatrix} \boldsymbol{d}_{11} & \boldsymbol{R}_1 \\ \boldsymbol{d}_{21} & \boldsymbol{R}_2 \end{vmatrix}$$

$$\boldsymbol{Z}_4 = \frac{1}{\boldsymbol{E}}\begin{vmatrix} \boldsymbol{R}_1 & \boldsymbol{e}_{12} \\ \boldsymbol{R}_2 & \boldsymbol{e}_{22} \end{vmatrix} \qquad \boldsymbol{Z}_6 = \frac{1}{\boldsymbol{E}}\begin{vmatrix} \boldsymbol{e}_{11} & \boldsymbol{R}_1 \\ \boldsymbol{e}_{21} & \boldsymbol{R}_2 \end{vmatrix}$$

$$\boldsymbol{Z}_3 = \boldsymbol{Z}_5 - \boldsymbol{Z}_6 \qquad \boldsymbol{Z}_1 = \boldsymbol{Z}_2 + \boldsymbol{Z}_3 - \boldsymbol{Z}_4$$

$$\boldsymbol{D} = \begin{vmatrix} \boldsymbol{d}_{11} & \boldsymbol{d}_{12} \\ \boldsymbol{d}_{21} & \boldsymbol{d}_{22} \end{vmatrix} = \begin{vmatrix} e^{j\phi_1} - 1 & e^{j\gamma_1} - 1 \\ e^{j\phi_2} - 1 & e^{j\gamma_2} - 1 \end{vmatrix}$$

$$\boldsymbol{E} = \begin{vmatrix} \boldsymbol{e}_{11} & \boldsymbol{e}_{12} \\ \boldsymbol{e}_{21} & \boldsymbol{e}_{22} \end{vmatrix} = \begin{vmatrix} e^{j\Psi_1} - 1 & e^{j\gamma_1} - 1 \\ e^{j\Psi_2} - 1 & e^{j\gamma_2} - 1 \end{vmatrix}$$

其中，

$$\boldsymbol{R}_k = |\boldsymbol{R}_k| e^{j\delta_k}$$

比较图 8.21 和图 8.25 可以看出，$|\boldsymbol{Z}_1| = L_1$，$|\boldsymbol{Z}_2| = L_2$，$|\boldsymbol{Z}_3| = L_3$，$|\boldsymbol{Z}_4| = L_4$。

① 参见 A G Erdman 和 G N.Sandor，出处同上，第 530 ~ 532 页。

对 R_k、ϕ_k、γ_k 和 Ψ_k，$k=1,2$ 的值进行假设。在这种情况下能确定六个连杆的长度和方向角 α、β、σ 和 θ。首先创建如下函数，用来成对计算 Z_2 和 Z_5，Z_4 和 Z_6。

```
function z = FourBarSynth(phi,gama,R)
coeff = [exp(j * phi(1)) - 1 exp(j * gama(1)) - 1;exp(j * phi(2)) - 1 exp(j * gama(2)) - 1];
D = det(coeff);
z1 = det([R(1)coeff(1,2);R(2)coeff(2,2)])/D;
z2 = det([coeff(1,1)R(1);coeff(2,1)R(2)])/D;
z = [z1 z2];
```

下面说明这些结果。

例 8.21 四连杆机构的综合

设如下数值：

$\phi_1=340°$	$\gamma_1=-48°$	$\delta_1=-31°$
$\phi_2=325°$	$\gamma_2=9°$	$\delta_2=-15°$
$\Psi_1=31°$	$\lvert P_1\rvert=2.7$	
$\Psi_2=81°$	$\lvert P_2\rvert=3.9$	

程序清单是：

```
c = pi/180;phi = [340.0 325.0] * c;gama = [ -48 9.0] * c;
R = [2.7 * exp( -31.0 * j * c)3.9 * exp( -15.0 * j * c)];
z25 = FourBarSynth(phi,gama,R);
disp(['Z2 = 'num2str(abs(z25(1)))'theta = '...
           num2str(angle(z25(1))/c)'degrees'])
disp(['Z5 = 'num2str(abs(z25(2)))'alpha = '...
           num2str(angle(z25(2))/c)'.degrees'])
psi = [31 80] * c;
z46 = FourBarSynth(psi,gama,R);
disp(['Z4 = 'num2str(abs(z46(1)))'sigma = '...
         num2str(angle(z46(1))/c)'degrees'])
disp(['Z6 = 'num2str(abs(z46(2)))'beta = '...
         num2str(angle(z46(2))/c)'degrees'])
Z3 = z25(2) - z46(2);
Z1 = z25(1) + Z3 - z46(1);
disp('Z3 = 'num2str(abs(Z3))])
disp('Z1 = 'num2str(abs(Z1))])
```

执行程序，在 MATLAB 命令窗口中显示如下信息：

```
Z2 = 6.7143  theta = 90.5276 degrees
Z5 = 1.2378  alpha = 24.8126 degrees
Z4 = 3.1762  sigma = -143.832 degrees
Z6 = 1.2774  beta = 95.0336 degrees
Z3 = 1.447
Z1 = 8.6814
```

这些结果显示在图 8.26 中，用来生成图形的程序留在练习 8.7 中完成。

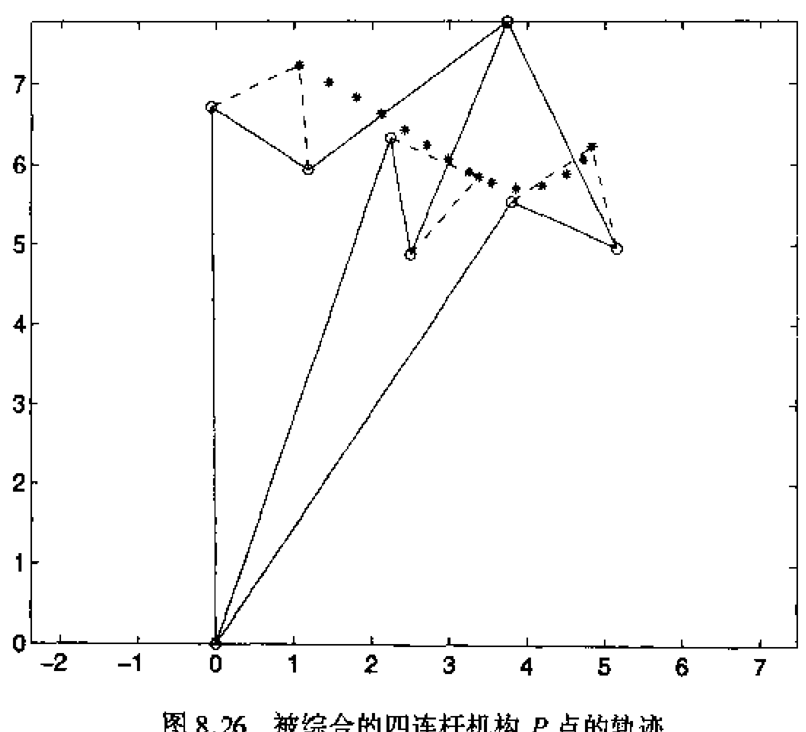


图 8.26　被综合的四连杆机构 P 点的轨迹

8.5　凸轮轮廓及其综合

8.5.1　凸轮位移

凸轮是把一种运动转化为另一种运动的装置。凸轮有一个向外弯曲的或是一个向内凹进的表面并与从动件一起实现运动形式的转换。凸轮通常是旋转运动形式，旋转运动可被转化成振动、直线运动或是两者的结合。应该考虑旋转运动的情况下对于从动件的位移轮廓。有两种类型的从动件，一种是表面盘形从动件，另一种是偏轴滚子从动件。综合的目的是确定凸轮轮廓和给定半径的剪床坐标，这些都是用来生成凸轮轮廓的。

定义一个凸轮基圆 r_b 作为最小的圆周半径，它与凸轮表面相切，而且和凸轮的转轴同心。从动件的运动方程如下：

$$L(\varphi) = r_b + s(\varphi)$$

对于旋转运动有：

$$s(\varphi) = h\left(\frac{\varphi}{\beta} - \frac{1}{2\pi}\sin(2\pi\varphi/\beta)\right) \qquad 0 \leqslant \varphi \leqslant \beta$$

$$s(\varphi) = h - h\left(\frac{\varphi - \beta}{\beta} - \frac{1}{2\pi}\sin(2\pi(\varphi - \beta)/\beta)\right) \qquad \beta < \varphi < 2\beta$$

$$s(\varphi) = 0 \qquad 2\beta \leqslant \varphi \leqslant 2\pi$$

上式是从动件的位移，h 是从动件的最大位移，并且 $0 \leqslant \beta \leqslant \pi$。

如果假设凸轮的旋转速度 $\omega = \mathrm{d}\varphi/\mathrm{d}t$ 是个常量，则速度 v、加速度 a 和瞬时加速度 j（加速度对时间求导）分别如下：

速度：

$$v(\varphi) = \frac{\omega h}{\beta}(1 - \cos(2\pi\varphi/\beta)) \qquad 0 \leqslant \varphi \leqslant \beta$$

$$v(\varphi) = -\frac{\omega h}{\beta}(1 - \cos(2\pi(\varphi - \beta)/\beta)) \qquad \beta \leqslant \varphi \leqslant 2\beta$$

$$v(\varphi) = 0 \qquad 2\beta \leqslant \varphi \leqslant 2\pi$$

加速度：

$$a(\varphi) = \frac{2\pi\omega^2 h}{\beta^2}\sin(2\pi\varphi/\beta)) \qquad 0 \leqslant \varphi \leqslant \beta$$

$$a(\varphi) = -\frac{2\pi\omega^2 h}{\beta^2}\sin(2\pi(\varphi - \beta)/\beta)) \qquad \beta \leqslant \varphi \leqslant 2\beta$$

$$a(\varphi) = 0 \qquad 2\beta \leqslant \varphi \leqslant 2\pi$$

瞬时加速度：

$$j(\varphi) = \frac{4\pi^2\omega^3 h}{\beta^3}\cos(2\pi\varphi/\beta)) \qquad 0 \leqslant \varphi \leqslant \beta$$

$$j(\varphi) = -\frac{4\pi^2\omega^3 h}{\beta^3}\cos(2\pi(\varphi - \beta)/\beta)) \qquad \beta \leqslant \varphi \leqslant 2\beta$$

$$j(\varphi) = 0 \qquad 2\beta \leqslant \varphi \leqslant 2\pi$$

定义无量纲位移 $S = s/h$、无量纲速度 $V = v/\omega h$、无量纲加速度 $A = a/h\omega^2$ 和无量纲瞬时加速度 $J = j/h\omega^3$。若 $\beta = 60°$，则利用如下程序可获得如图 8.27 所示的结果。

```
beta = 60 * pi/180;
phi = linspace(0,beta,40);
phi2 = [beta + phi];
ph = [phi phi2] * 180/pi;
arg = 2 * pi * phi/beta;
arg2 = 2 * pi * (phi2 - beta)/beta;
s = [phi/beta - sin(arg)/2/pi 1 - (arg2 - sin(arg2))/2/pi];
v = [1 - cos(arg))/beta - (1 - cos(arg2))/beta];
a = [2 * pi/beta^2 * sin(arg)2 * pi/beta^2 * sin(arg2)];
j = [4 * pi^2/beta^3 * cos(arg)4 * pi^2/beta^3 * cos(arg2)];subplot(2,2,1)
plot(ph,s,'k')
xlabel('Cam angle(degrees)')
ylabel('Displacement(S)')
g = axis;g(2) = 120;axis(g)
subplot(2,2,2)
plot(ph,v,'k',[0 120],[0 0],'k--')
xlabel('Cam angle(degrees)')
ylabel('Velocity(V)')
g = axis;g(2) = 120;axis(g)
subplot(2,2,3)
plot(ph,a,'k',[0 120],[0 0],'k--')
xlabel('Cam angle(degress)')
ylabel('Acceleration(A)')
g = axis;
g(2) = 120;
axis(g)
subplot(2,2,4)
plot(ph,j,'k',[0 120],[0 0],'k--')
```

```
xlabel('Cam angle(degrees)')
ylabel('Jerk(J)')
g=axis;
g(2)=120;
axis(g)
```

研究两个特殊的凸轮结构:一个是直线表面盘形从动件,另一个是偏轴滚子从动件。

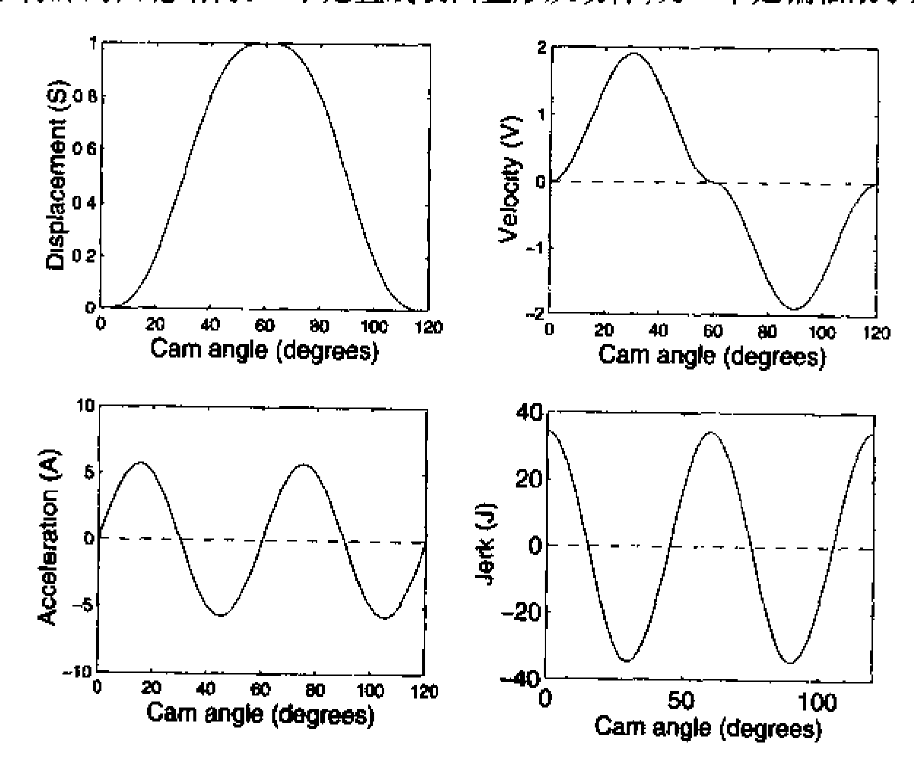


图 8.27　摆线凸轮轮廓的标准位移、速度、加速度和瞬时加速度

8.5.2 表面盘形从动件

参考图 8.28 得到如下关系[①]:在(x,y)坐标系中,凸轮轮廓的坐标为 R_x 和 R_y,刀具的坐标为 C_x 和 C_y:

$$R_x = R\cos(\theta+\varphi) \qquad R_y = R\sin(\theta+\varphi)$$
$$C_x = C\cos(\gamma+\varphi) \qquad C_y = C\sin(\gamma+\varphi)$$

其中,

$$R = \frac{L}{\cos\theta} \qquad \theta = \arctan\left(\frac{1}{L}\frac{\mathrm{d}L}{\mathrm{d}\varphi}\right)$$
$$C = \frac{L+r_c}{\cos\gamma} \qquad \gamma = \arctan\left(\frac{\mathrm{d}L/\mathrm{d}\varphi}{L+r_c}\right)$$

① 参见 A G Erdman 和 A G Erdman G.N Sandor,出处同上,第 385–387 页和 H.A Rothbart, *Cams: Design, Dynamics, Accuracy*, John Wiley & Sons, New York, 1956 年,第 122–124 页。

r_c 是刀具的半径,且 $dL/d\varphi = V(\varphi)/\omega$。

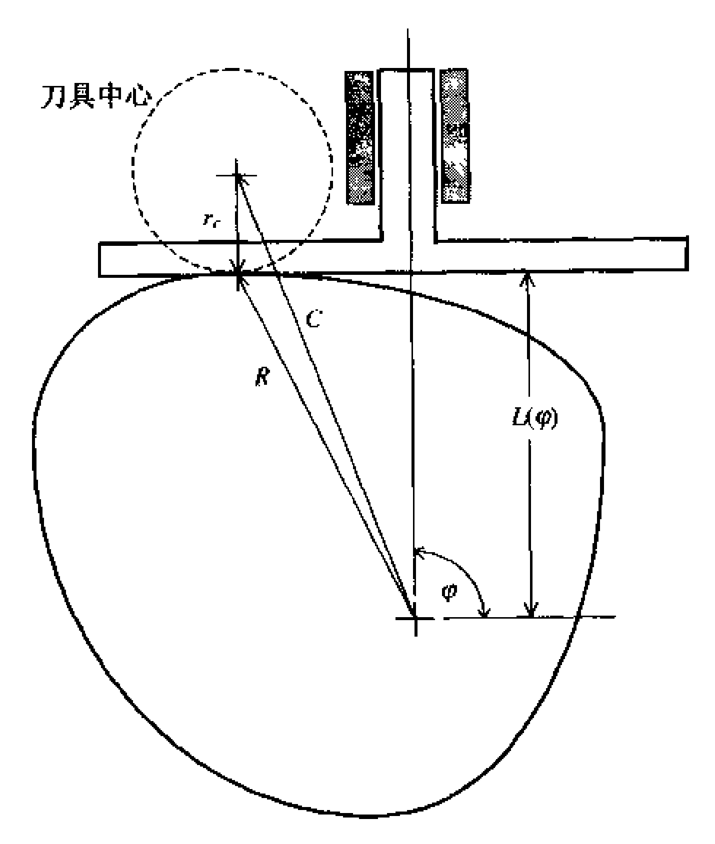


图 8.28 带转换平面从动件的凸轮

为了显示如图 8.29 所示的结果,创建函数 CamProfile, 用来计算 $L(\varphi)$和 $dL/d\varphi$,创建函数 ContourFlat, 用来计算 R_x, R_y, C_x, C_y。

```
function[L,dLdphi] = CamProfile(phi,rb,h,beta)
arg = 2 * pi * phi/beta;
L = rb + h * (phi/beta - sin(arg)/2/pi);
dLdphi = (h/beta) * (1 - cos(arg));
L = [L fliplr(L)];
dLdphi = [dLdphi - dLdphi];
```

函数 ContourFlat 是:

```
function[Rx,Ry,Cx,Cy] = ContourFlat(phi,rb,h,beta,rc)
[L,dLdphi] = CamProfile(phi,rb,h,beta);
theta = atan2(dLdphi,L);
R = L./cos(theta);
ph = [phi beta + phi];
Ry = R. * sin(theta + ph);
Rx = R  * cos(theta + ph);
gama = atan(dLdphi./(L + rc));
C = (L + rc)/cos(gama);
Cy = C  * sin(gama + ph);
Cx = C  * cos(gama + ph);
```

若令 $\beta = 60°, r_b = 3.0$ 和 $h = 0.5$,则程序清单是:

```
beta = 60 * pi/180; rb = 3; h = 0.5; rc = 0.5; n = 23;
phi = linspace(0,beta,n);
ph = [phi beta + phi];
[Rx,Ry,Cx,Cy] = ContourFlat(phi,rb,h,beta,rc);
ang = linspace(2 * beta,2 * pi,40);
plot(Rx,Ry,'k',rb * cos(ang),rb * sin(ang),'k',0,0,'k+',Cx(1:5:2 * n),Cy(1:5:2 * n),'k+')
axis equal
phd = linspace(0,2 * pi,50);
[x,phx] = meshgrid(Cx(1:5:2 * n),phd);
y = meshgrid(Cy(1:5:2 * n),phd);
hold on
plot(x + rc. * cos(phx),y + rc. * sin(phx),'k--')
title('Cam contour for cycloidal motion of flat-face follower')
```

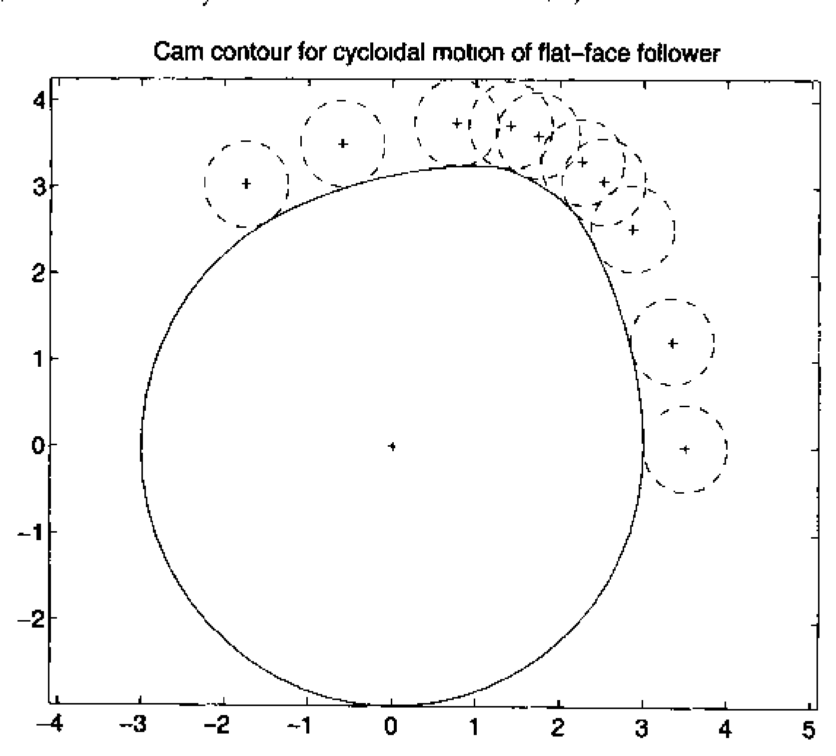


图 8.29　相对于平面从动件摆线运动的凸轮轮廓及其刀具位置

8.5.3　偏轴滚子从动件

参考图 8.30 得到如下关系。[1]凸轮轮廓在(x,y)的坐标是 R_x 和 R_y，刀具的坐标是 C_x 和 C_y：

$$R_x = R\cos(\Psi + \varphi + \gamma) \qquad R_y = R\sin(\Psi + \varphi + \gamma)$$

$$C_x = C\cos(\Psi + \varphi + \delta) \qquad C_y = C\sin(\Psi + \varphi + \delta)$$

其中，

$$R^2 = (F - r_f\cos\alpha)^2 + r_f^2\cos^2\alpha \quad \Psi = \arctan(m/L)$$

$$C^2 = c_x^2 + c_y^2 \qquad \alpha = \arctan\left(\frac{L\mathrm{d}L/\mathrm{d}\varphi}{F^2 - m\mathrm{d}L/\mathrm{d}\varphi}\right)$$

① 参见 A G Erdman 和 G.N.Sandor，出处同上，第 389 ~ 393 页和 H.A Rothbart，出处同上，第 120 ~ 122 页。

$$c_x = F + (r_c - r_f)\cos\alpha \quad \gamma = \arctan\left(\frac{F - r_f\cos\alpha}{r_f\sin\alpha}\right)$$

$$c_y = (r_c - r_f)\sin\alpha \qquad \delta = \arctan(c_y/c_x)$$

$$F^2 = m^2 + L^2$$

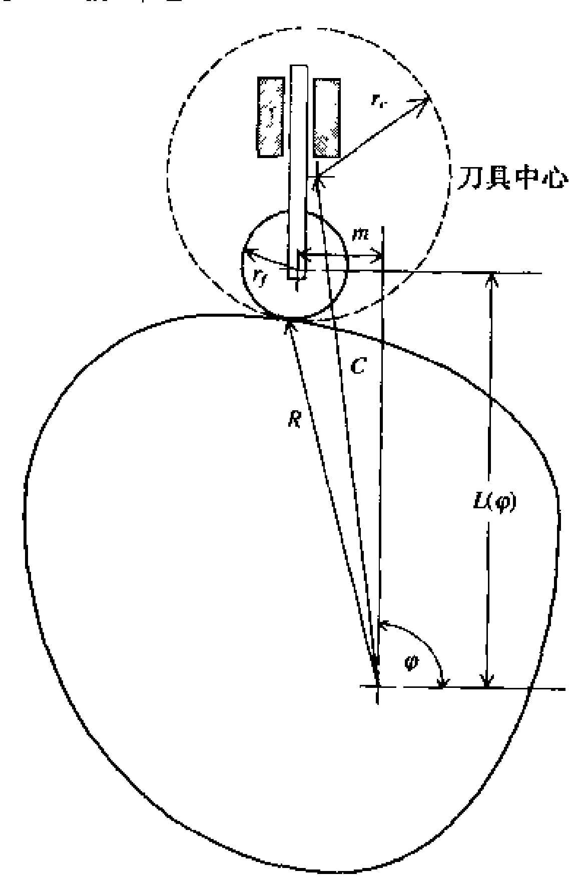


图 8.30 有偏轴滚子从动件的凸轮

凸轮的基圆半径是：

$$L(0) = r_b = \sqrt{R_x^2(\varphi = 0) + R_y^2(\varphi = 0)}$$

从 $\varphi = 2\beta + \Delta$ 开始，其中，

$$\Delta = \arctan\left(\frac{R_y(\varphi = 0)}{R_x(\varphi = 0)}\right)$$

为了显示这些结果，首先创建函数 ContourRoller 来计算 R_x，R_y，C_x，C_y：

```
function[Rx,Ry,Cx,Cy] = ContourRoller(phi,rb,h,beta,rc,m,rf)
[L,dLdphi] = CamProfile(phi,rb,h,beta);
F2 = m^2 + L.^2;
F = sqrt(f2);
psi = atan2(m,L);
alpha = atan2(L. * dLdphi,F2 - m * dLdphi);
gamma = atan2(rf * sin(alpha),F - rf * cos(alpha));
ph = [phi beta + phi];
R = sqrt((F - rf * cos(alpha)).^2 + (rf * sin(alpha)).^2);
Ry = R  * sin(psi + gamma + ph);
Rx = R. * cos(psi + gamma + ph);
cx = F + (rc - rf) * cos(alpha);
```

```
cy = (rc - rf) * sin(alpha);
delta = atan2(cy,cx);
C = sqrt(cx.^2 + cy.^2);
Cy = C. * sin(psi + delta + ph);
Cx = C. * cos(psi + delta + ph);
```

若令 $\beta = 60°, r_b = 3.0, h = 0.5, r_c = 0.5, r_f = 0.375, m = 0.375$,则程序清单是:

```
beta = 60 * pi/180; rb = 3; h = 0.5; rc = 0.5; rf = 0.375; m = .375; n = 23;
phi = linspace(0,beta,n);
ph = [phi beta + phi];
[Rx,Ry,Rx,Ry] = ContourRoller(0,rb,h,beta,rc,m,rf);
rb = sqrt(Rx(1)^2 + Ry(1)^2);
delta = atan2(Ry(1),Rx(1));
[Rx,Ry,Cx,Cy] = ContourRoller(phi,rb,h,beta,rc,m,rf);
ang = linspace(2 * beta + delta,2 * pi + delta,40);
plot(Rx,Ry,'k',Rx(1) * cos(ang),Rx(1) * sin(ang),'k',0,0,'k+',Cx(1:5:2 * n),..
     Cy(1:5:2 * n),'k+')
axis equal
phd = linspace(0,2 * pi,50);
[x,phx] = meshgrid(Cx(1:5:2 * n),phd);
y = meshgrid(Cy(1:5:2 * n),phd);
hold on
plot(x + rc. * cos(phx),y + rc. * sin(phx),'k--')
title('Cam contour for cycloidal motion of an offset roller follower')
```

程序执行结果如图 8.31 所示。

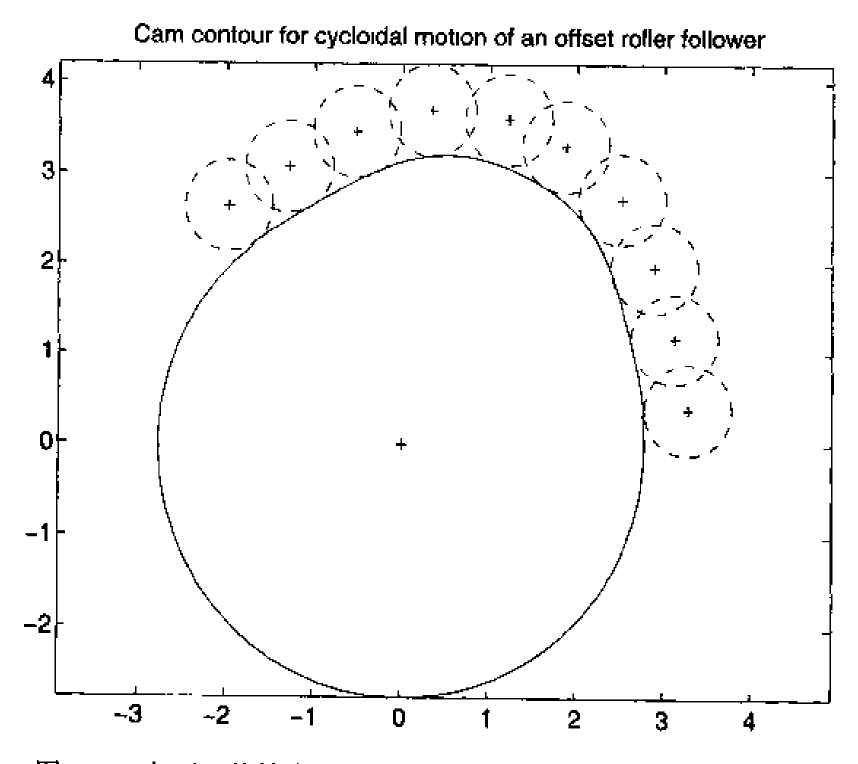


图 8.31　相对于偏轴滚子从动件摆线运动的凸轮轮廓及其刀具坐标

8.5.4 凸轮的曲率半径

凸轮轮廓的曲率半径如下给出：

$$\rho=\frac{[(L(\varphi))^2+(\mathrm{d}L(\varphi)/\mathrm{d}\varphi)^2]^{3/2}}{(L(\varphi))^2+2(\mathrm{d}L(\varphi)/\mathrm{d}\varphi)^2-L(\varphi)\mathrm{d}^2L(\varphi)/\mathrm{d}\varphi^2}$$

凸轮轮廓应该是这样的：从动件的曲率半径总是大于凸轮轮廓的最小曲率半径，有意义的是最小曲率半径。使用第 8.5.1 节中对无量纲位移、速度、加速度和瞬时加速度的定义，曲率半径可表示如下：

$$\rho/h=\frac{[(r_b/h+S)^2+V^2]^{3/2}}{(r_b/h+S)^2+2V^2-(r_b/h+S)A}$$

为了确定最小的曲率半径(无量纲的)，利用曲率半径与 β 的对称关系创建函数 CamCurvature。在 $0\leqslant\varphi\leqslant\beta$ 区间是有效的。

```
function RadiusCurve = CamCurvature(phi,beta,rbh)
arg = 2 * pi * phi/beta;
S = phi/beta - sin(arg)/2/pi;
V = (1 - cos(arg))/beta;
A = 2 * pi/beta^2 * sin(arg);
RadiusCurve = ((rbh + S)^2 + V^2)^1.5/((rbh + S)^2 + 2 * V^2 - (rbh + S) * A);
```

那么对于 r_b/h 和 β 的任意值，程序清单是：

```
rbh = input('Enter ratio rb/h:');
beta = input('Enter angle beta(degrees):') * pi/180;
options = optimset('display','off');
phimin = fminbnd('CamCurvature',0,beta,options,beta,rbh);
rmin = CamCurvature(phimin,beta,rbh);
disp(['When beta = 'num2str(beta * 180/pi)'degrees and rb/h = '...
      num2str(rbh)'the minimum radius of curvature for a'])
disp (['cycloidal cam profile is = 'num2str(rmin)'h,which occurs at'...
      num2str(phimin * 180/pi)'degrees.'])
```

程序执行后，在 MATLAB 命令窗口中显示如下信息：

```
Enter ratio rb/h:4
Enter angle beta(degrees):80
When beta = 80 degrees and rb/h = 4 the minimum radius of curvature for a
cycloidal cam profile is = 2.9777h,which occurs at 58.8421 degrees.
```

其中用户输入值是 4 和 80。

8.6 滑动轴承

考虑如图 8.32 所示的短颈轴承。如果 $c_d=(d_h-d_j)$，那么径向间隔 $c_r=c_d/2$ 是偏心距 e 的最大值，偏心率被定义为 $\varepsilon=e/c_r$。然而，实际上这个值通常由下面的经验公式获得：

$$\varepsilon\rightarrow\varepsilon_x=0.213\,94+0.385\,17\log_{10}O_N-0.000\,8(O_N-60)$$

其中 O_N 是一个载荷系数，或称为 Ocvirk 数，由下式给出：

$$O_N = \frac{P}{nLd\eta}\left(\frac{d}{L}\right)^2\left(\frac{c_d}{d}\right)^2 = 4\pi K_\varepsilon$$

其中 η 是油的绝对粘度，n 是轴颈旋转速度，以每秒运行的圈数来记。设计目标是保持 $O_N <$ 30。

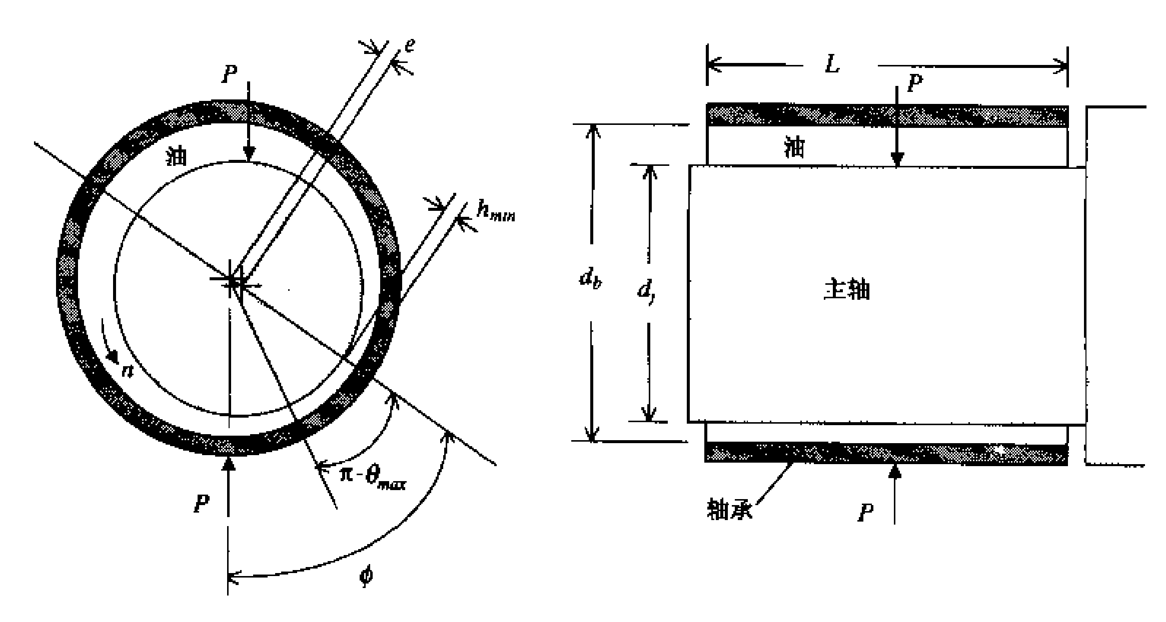


图 8.32　关于滑动轴承的术语

最小的油膜厚度如下给出：

$$h_{min} = c_r(1-\varepsilon_x)$$

为了最大程度减小表面接触的机会，它应该是比轴承和轴颈表面光洁度大 2 到 3 倍的系数。

轴承能承受的载荷是 ：

$$P = 4\pi K_\varepsilon \eta n d L^3 / c_d^2$$

该转动轴转矩为：

$$T_r = \frac{\pi^2 d^3 L n \eta}{c_d\sqrt{(1-\varepsilon_x^2)}} + 0.5P\varepsilon_x c_d \sin\phi$$

其中在等号右边的第一项是稳定转矩，而：

$$\phi = \arctan\left(\frac{\pi\sqrt{1-\varepsilon_x^2}}{4\varepsilon_x}\right)$$

轴颈和轴承之间的摩擦系数是：

$$\mu = \frac{2T_r}{Pd}$$

油的华氏温度的增加值如下给出：

$$\Delta T = \frac{2\pi n T_r}{6\,600\rho c_p Q_H}$$

其中，

$$Q_H = \pi d n l \varepsilon_x c_d / 2$$

它是油的流量，单位是 in^3/s；润滑剂的热容量为 C_p；ρ 是它的密度。对于润滑油，有近似值 $c_p = 0.48$ Btu/(lb. - ℉)和 $\rho = 0.031 lb/in^3$。假设温度升高是从入口油的温度 T_{in} 开始的。因为 η

(见下面)和 T_r 是温度的函数,所以这个方程需要迭代求解。使用 c_p 和 ρ,为下式:

$$\Delta T = \frac{0.0640 n T_r}{Q_H}$$

油的绝对粘度通过如下关系可以在 10% 之内估计,单位是雷恩(lb-s/in²):

$$\eta = 10^{C-7}$$

其中,

$$C = 10^{A_j - B_j \log_{10} T_o}$$

其中 $T_o = 255.2 + 5/9 T$ K, T 是油的华氏温度, A_j 和 B_j 在表 8.6 中给出,是油的 SAE 值的函数。

表 8.6 用于确定 η 的常数

SAE 数	j	A_j	B_j
10	1	9 1209	3 5605
20	2	9.1067	3.5385
30	3	8.9939	3 4777
40	4	8 9133	3 4292
50	5	8 5194	3.2621
60	6	8.3666	3 1884

在用例子说明这些结果之前,创建 3 个函数:

函数 AbsViscosity 用于计算绝对粘度 η

函数 Ocvirk 用于计算 Ocvirk 数 O_N

函数 TempRise 用于计算温升 ΔT

函数 AbsViscosity 如下:

```
function absvis = AbsViscosity(tempF,SAE)
AandB = [9.1209 3.5605;9.1067 3.5385; 8.9939 3.4777;8.9133 3.4292;...
        8.5194 3.2621;8.3666 3.1884];
SAEn = SAE/10;
absvis = 10^(10^(AandB(SAEn,1)-AandB(SAEn,2) * log10(255.2 + 5/9 * tempF))-7);
```

函数 Ocvirk 如下:

```
function[ex,ocv] = Ocvirk(vis,L,d,cd,n,P)
ocv = 60 * P/n/L/d/vis * (cd/L)^2;
ex = 0.21394 + 0.38517 * log10(cov) - 0.0008 * (ocv - 60));
```

函数 TimpRise 如下:

```
function deltaT = TempRise(temp,n,d,cd,ex,L,P,SAE)
QH = pi * d * (n/60) * L * ex * cd/2
phi = atan(pi * sqrt(1 - ex^2)/4/ex);
vis = AbsViscosity(temp,SAE);
Tr = pi^2 * d^3 * L * (n/60) * vis/cd/sqrt(1 - ex^2) + 0.5 * P * ex * cd * sin(phi);
deltaT = (0.0640) * Tr * (n/60)/QH;
```

例 8.22　滑动轴承的载荷容量

考虑如下滑动轴承的指标：

油的入口温度 = 120°F
SAE 30 油
轴转速 = 3 000 r/m
轴直径 = 2.0 in
主轴长 = 1.1 in
环向间隙 = 0.001 3 in

当 Ocvirk 数低于 25 时，确定轴承能承受的最大载荷，以及相应的温升、最小的油膜薄片厚度和绝对粘度值。

首先创建函数 JournalLoad，对给定载荷计算油的温升、绝对粘度和 Ocvirk 数。fzero 函数使用该函数来求 Ocvirk 数等于 25 时的载荷值。

```
function[Omax,absvis,ex,temp] = JournalLoad(P,Tin,SAE,n,d,cd,L)
temp = Tin;Told = 0;
while abs(temp-Told) > 0.1
  Told = temp;
  absvis = AbsViscosity(temp,SAE);
  ex = Ocvirk(absvis,L,d,cd,n,P);
  deltaT = TempRise(temp,n,d,cd,ex,L,P,SAE);
  temp = Tin + deltaT;
end
absvis = AbsViscosity(temp,SAE);
[ex,ocv] = Ocvirk(absvis,L,d,cd,n,P);
Omax = 25 - ocv;
```

程序清单如下：

```
Tin = 120;SAE = 30;n = 3000;d = 2;cd = .0026;L = 1.1;
options = optimset('display','off');
P = fzero('JournalLoad',1000,options,Tin,SAE,n,d,cd,L);
[Omax,absvis,ex,temp] = JournalLoad(P,Tin,SAE,n,d,cd,L);
disp(['Maximum load = 'num2str(P)'lb'])
disp(['Temperature rise = 'num2str(temp-Tin)'degrees F'])
disp(['Absolute viscosity = 'num2str(absvis)'reyn'])
disp(['Minimum oil film thickness = 'num2str(0.5 * cd * (1 - ex))'inches'])
```

执行程序后，在 MATLAB 命令窗口中显示如下内容：

```
Maximum load = 1078.3427 lb.
Temperature rise = 58.6577 degrees F
Absolute viscosity = 2.1907e-006 reyn
Minimum oil film thickness = 0.0002855 inches
```

8.7 PDE 工具箱和薄板凹槽处的应力集中系数

计算 MATLAB 的 PDE(偏微分方程)工具箱[①] 和它的 GUI(图形用户界面)一起用来分析几类二维场方程。一类是平面应力问题,平面应力问题是一种近似方法,用于确定任意形状和厚度 h 的平面薄板的应力、拉力和位移。假设应力(σ_Z)在 $z = \pm h/2$ 处垂直于平板的表面并且在平板表面的剪切应力是零,则其他应力独立于 z。

使用工具箱和 GUI 在矩形钢板上确定集中应力系数。在平板两边分别开出两个对称于中点的半圆形凹槽,平板的形状和说明如图 8.33 所示。平板在左端固定,在平板的右端施加沿 x 轴方向上的力,每单位长度上力的大小为 $F_x = 20$ N/cm。如果平板的厚度是 0.2 cm,那么在 x 轴正方向上的边缘应力是 100 N/cm²。钢板的弹性模量是 200x10⁵ N/cm²。

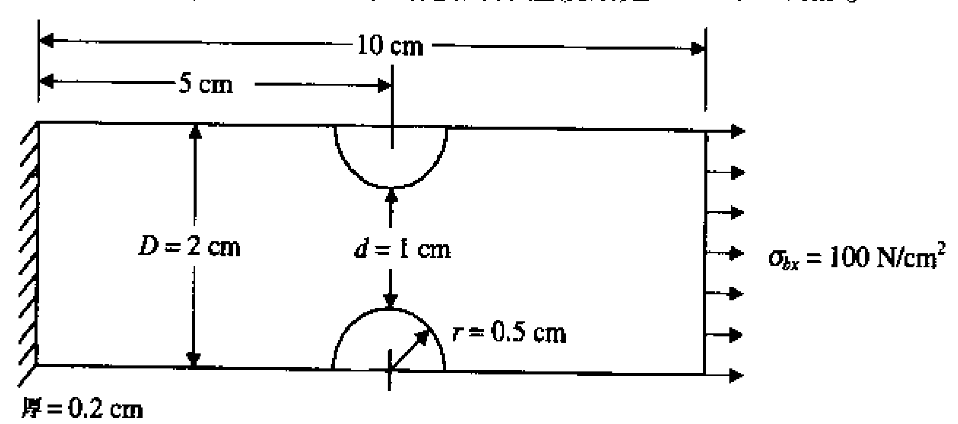


图 8.33 拉伸作用下的开槽平板

PDE 工具箱要求给出每一个边的边界条件,在平板应力问题中通常需要考虑两种类型的边界条件:(1)位移是零或是给定值,(2)表面应力是零或是给定值。位移的给定值在 GUI 中描述为第立克特(Dirichlet)边界条件,而应力在的给定值描述为纽曼(Neumann)边界条件。

在(x,y)坐标系中的任意方向上,研究作用在边缘上的两个应力。使用 GUI 注释,如果作用在 x 方向上的应力分量是 g_1,作用在 y 方向上应力分量是 g_2,则:

$$g_1 = n_x\sigma_{bx} + n_y\sigma_{bxy}$$

$$g_2 = n_y\sigma_{by} + n_x\sigma_{bxy}$$

其中,σ_{bx}是 x 方向上的边界应力,σ_{by}是 y 方向上的边界应力,σ_{bxy}是边界剪切应力。另外,n_x和n_y分别是 x 轴和 y 轴的方向余弦。这样,在与 x 轴平行的边界上,$n_x = 0$, $n_y = \pm 1$(+1 表示与 y 轴同向),规定表面应力为:

$$g_1 = \pm \sigma_{bxy}$$

$$g_2 = \pm \sigma_{by}$$

如果边缘部分的剪切应力是零,即 $\sigma_{bxy} = 0$,则 $g_1 = 0$;相反,如果法向应力 $\sigma_{by} = 0$,则 $g_2 = 0$。如果边缘部分无应力,则 $g_1 = g_2 = 0$。

当处理弯曲边缘(圆弧或椭圆弧)时,可以规定边界条件如下:

① PDE 工具箱在流动方面的可视化应用参见第 11 3.3 节,在传热方面的应用参见第 12 1 节。

$$g_1 = N * nx$$

$$g_2 = N * ny$$

其中，nx 和 ny 被 GUI 分别表示为相对于 x 轴和 y 轴的方向余弦，并且 N 是一个数字值，用于表示弯曲边界的法向应力值。

使用 PDE GUI 来确定应力、拉力和位移的过程有如下几步：

1. 设置画图面积；
2. 选择合适的 PDE——在这种情况下应选平面应力；
3. 画出（或创建）二维形状图；
4. 规定边界条件；
5. 选择物理量；
6. 生成格状图；
7. 求解；
8. 显示结果；
9. 如果需要进一步分析，则把 GUI 生成的数组输出到 MATLAB 的命令窗口中去。

下面给出这些步骤的实现细节。在这一过程中任何一个步骤都能返回到前面已完成的步骤并进行改变，那么前面已完成的步骤也必须能再次被执行。为了访问 GUI，在 MATLAB 命令窗口中输入：

```
Pdetool
```

这是一个 MATLAB 函数。

设置画图面积　基于图 8.33 中对平板的说明，创建一块 12×4 单位的绘图面积，其中 1 单位 =1 cm。然后显示网格线，且通过点击 Snap 出现“整格移动”选项，此后多次点击 Options 菜单。第一次点击 Grid，出现检查记号（√）。然后在 Snap 上做同样的操作。接下来点击 Axis Limits。在菜单窗口中，输入 X 轴范围：

```
[0  12]
```

输入 y 轴范围：

```
[0  4]
```

然后点击 Apply 并 Close。先点击 Apply 确定 Auto boxes 是空的。最后，点击 Grid Spacing，输入 x 轴的线性分割为：

```
0:0.5:12
```

y 轴的线性分割为：

```
0:0.5:4
```

将该面积标记为空，确保 Auto boxes 也是空的。点击 Apply，然后点击 Done。

选择 PDE　为了选择 PDE，进入工具条的右端，点击指向描述符 Generic Scalar 旁的箭头。当 pdetool 打开后，该描述符就能显示出来，然后选择 Structural Mech 和 Plane Stress。

画出(创建)二维形状图　绘图程序采用积木方法创建平面图形,通过对矩形(正方形)、椭圆(圆)和多边形的任意组合进行填加和删除,从而完成绘图过程。每一个图元都能被放置在已创建区域的任何地方,并且能加入说明,但不能超过所画区域的边缘。十字光标的位置坐标在 GUI 窗口的右上角给出。点击绘图区域左上角的某种图标即可选中该图元,并将其绘制出来。根据选择方式不同,用鼠标键可按图元的顶角放置,也可按中心放置。然后按下鼠标键,光标移动到下一个位置并松开。Snap 选项可简化这种放置过程。

每个图元放置好以后,一个文字数字指示器显示在刚创建的图元中心,同时显示在输入数据窗口,输入数据窗口在绘图区的上端和 Set formula 注释区的右侧。约定是:*R* 代表矩形,*SQ* 代表正方形,*E* 代表椭圆,*C* 代表圆,*P* 代表多边形。一个图元被放置后它的文字数字指示器的符号设定为正号(+)。要从一个图中移走(或删除)一个图元,应进入到数据区域改变相应图元的符号为负号(-)。变化符号以后,按回车键。

通过创建图 8.33 所示的平面图来说明这一步。首先点击工具条打开一个矩形,然后将光标放置在坐标(1,1)处并点击,按住鼠标键拖动至坐标(11,3)处,放开鼠标键,就得到一个 10×2 的矩形。如果不想要此矩形,那么到 Edit 菜单选择 Clear 或在键盘上按下 Del 键加以清除。当放置了多个图元时,要首先选中被删除的图元,否则所有的图元都将被删除。被选中的图元有一个黑色的边框。

接下来选择中心的放置方式,将光标放置在(6,1)处画一个椭圆(圆),按下鼠标键,拖动光标直到出现一个直径为 1 的圆。因为坐标轴是不等的,所以在做图区域中出现一个椭圆,重复这一过程可画出中心为(6,3)的圆。最终结果如图 8.34 所示。

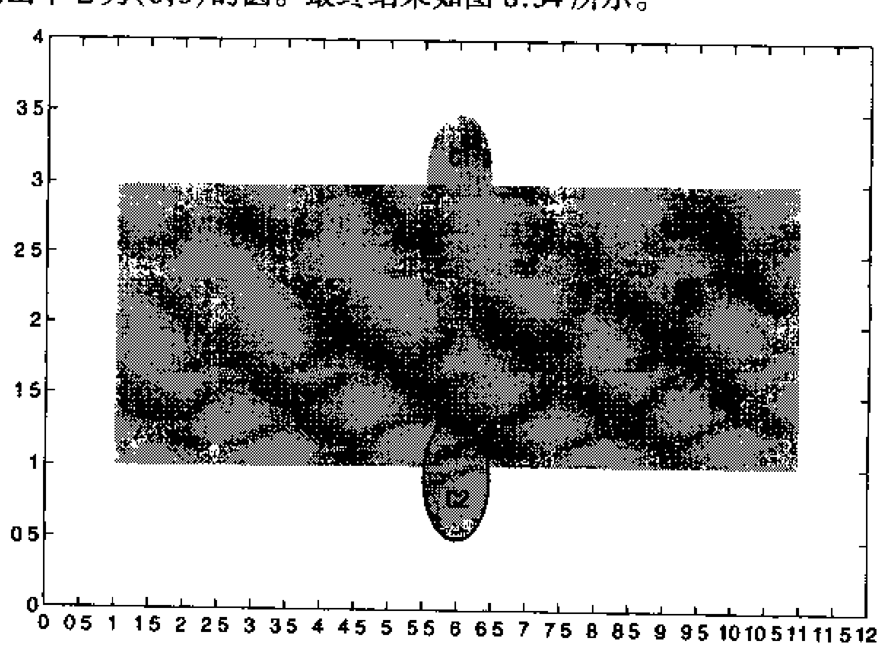


图 8.34　画出一个矩形和两个圆的结果

在 Set formula 区域中将:

$R1 + C1 + C2$

变为:

$R1 - C1 - C2$

并按回车键。

当不需要网格线时，可以将其关闭。

指定边界条件 在规定每一个边缘部分的边界条件以前，必须让GUI执行一系列的操作。首先进入Boundary菜单并选择Boundary Mode选项。操作结果如图8.35所示。将区域中的图形设置为边界选择模式。

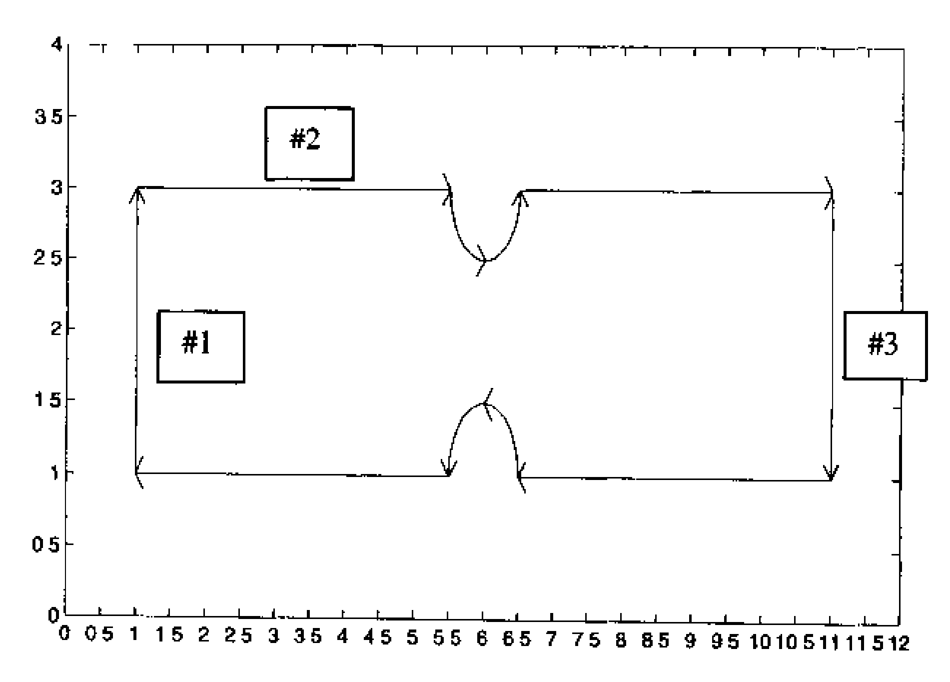


图8.35 设置成Boundary Mode操作的结果
（注意：图上的数字并不属于显示内容）

接下来的工作将会围绕每一个边界部分展开，边界部分开始于一个箭头的顶端，结束于另一个相邻箭头的顶端。按任意顺序确定每一部分的边界条件，建议采用顺时针方向或逆时针方向，最大限度保证边界不丢失。在段#1开始，沿图形边界顺时针旋转。

当点击段#1时，它由红变黑。然后双击这一部分，出现该部分的Boundary Conditions窗口。因为边缘是固定的，所以具有零位移。就像前面所提到的，位移是第立克特边界条件。这样，在Conditon type中选择Dirichlet。既然位移是零，可以使用默认的边界条件：$h_{11}=h_{22}=1$，$h_{12}=h_{21}=r_1=r_2=0$。位移标号出现在菜单窗口的上方，u代表两个向量分量：一个是x方向上的位移，另一个是y方向上的位移。这样r_1与x方向上的位移相关，r_2与y方向上的位移相关。点击OK按钮关闭边界条件窗口，线段的颜色仍恢复为红色。

因为剩余边缘部分可能无应力作用或有一个已知的应力作用（#3部分），所以都是纽曼边界条件。这样对于段#2和所有的无应力作用的边缘部分使用如下过程。再次双击该线段，显示Condition type选择Neumann，其他值都是默认值，即0。点击OK按钮、关闭边界条件窗口，此时线段为蓝色，除线段#3以外其余的线段都可以这样做。对于线段#3再选择Neumann。$g1$的输入值是100，代表100 N/cm^2。其余的默认值保持不变。

指定物理常量 为了指定平板的物理常量，进入PDE并选择PDE Specification选项。E值的输入为200e5，默认值$nu=0.3$（泊松比）和$rho=1$（质量密度）。因为是静态问题，所以不使用密度，然后点击OK按钮。

生成格状图 为了让系统生成格状图，从Mesh菜单中选择Initialize Mesh选项，结果如图8.36

所示。如果网格显示过份粗糙,那么回到 Mesh 菜单并选择 Refine Mesh 选项,结果如图 8.37 所示。每选择 Refine Mesh 选项一次,格状三角形就变小一次。

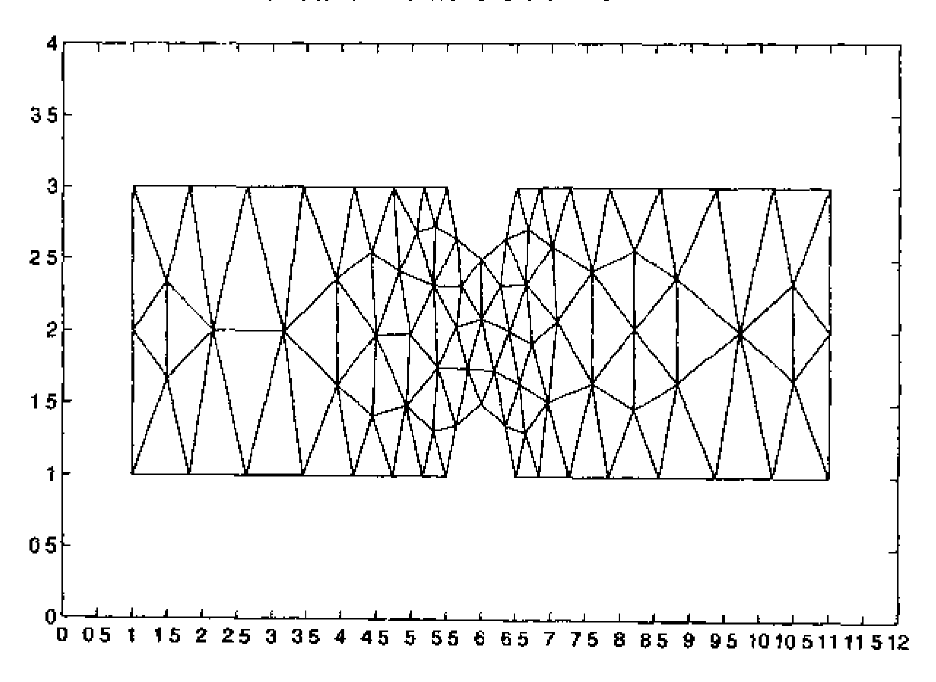


图 8.36 从 Initialize Mesh 选项得到的结果

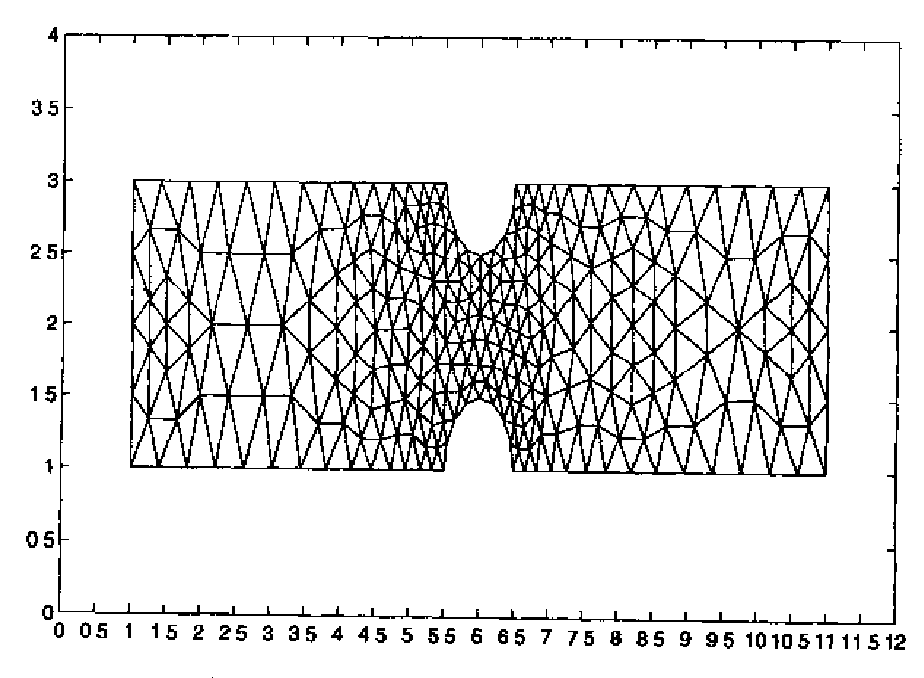


图 8.37 使用一次 Rifine Mesh 选项后的结果

求解 求解的方法是选择 Solve 菜单中的 Solve PDE 选项或者点击"="图标。

显示结果 若以一种形式显示多个结果,则最好对它们进行说明。进入 Plot 菜单并选择 Parameters 选项。σ_{xx}的分布见图 8.38,在 Plot type 菜单下选择 Color 选项和 Contour 选项,在 Property 的第一个可选框中选择 x stress 选项。

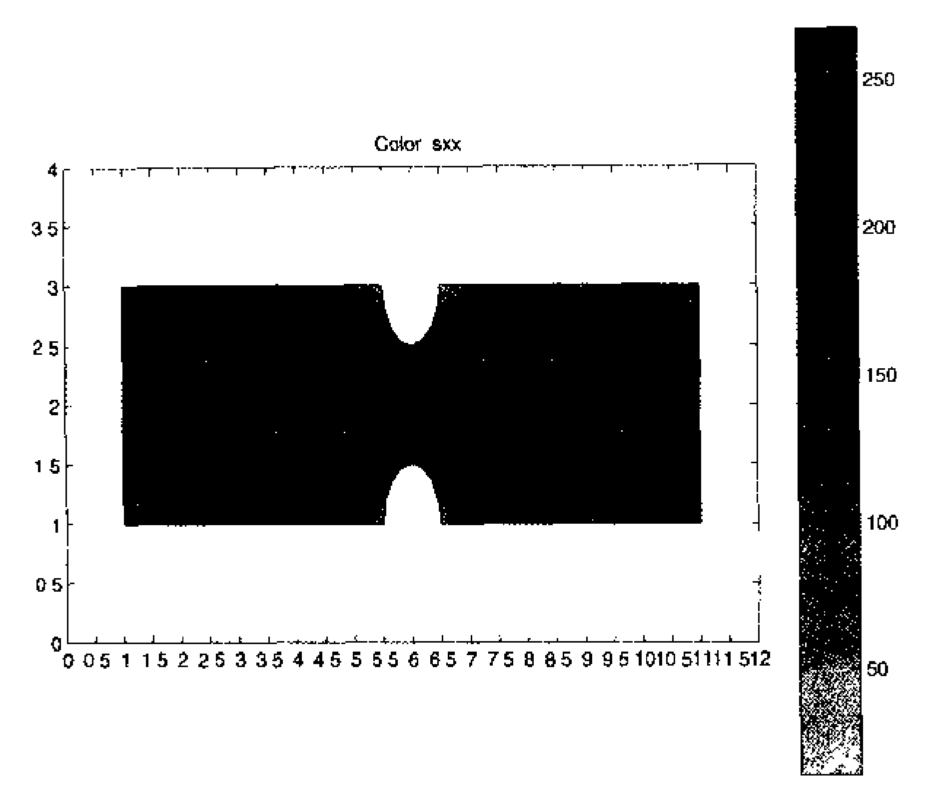


图8.38　σ_{xx}应力等高线和色棒

图8.34至图8.37被作为与插图封装后的二层压缩文件存储，该压缩文件作为可见图形用于MS Word文档中。这些可以按如下方式完成，即在File菜单上选择Print，在Device option数据区域中输入：

```
-deps2  -tiff
```

按Enter键。在Send to区域中选择File然后点击Save。这就打开了一个典型的Save窗口，以常用的方式选择路径名和文件名。

文件以".eps"作为扩展名。其他选项参见Print的帮助文件。

将图形转到Windows剪贴板中，使用：

```
-dmeta
```

然后选择Printer。

把由GUI生成的数组输出到MATLAB命令窗口中　为了确定应力集中系数，需要有应力σ_{xx}的最大值。所以，必须把适当的结果传送(输出)到MATLAB的命令窗口中去。在这个窗口里使用：

```
StressX = pdesmech(p,t,c,u,'tensor','sxx');
```

来获得σ_{xx}的数组，然后使用函数max寻找最大值。其中p给出了网格的坐标点，e描述了格状三角形的边缘，t描述角度，u是计算结果，参见initmesh的帮助文件。所有的这些数值都从GUI输出。在MATLAB命令窗口中使用的计算结果必须有一个，可以使用由MATLAB提供的、在上面的函数中以参数形式出现的变量名，如果它们和以前定义的全局变量冲突，则可以更名。同时，必须在pdesmech的参数表中更名。

进入Mesh菜单选择Export Mesh选项，将p和t输出到MATLAB命令窗口中，这时显示变量名，点击OK按钮。如果进入MATLAB命令窗口敲入p(并回车)，则会出现数组(因为数组可

能很大,所以不提倡这样做)。此时可对多个变量更名,然后点击 OK 按钮。

进入 PDE 菜单并选择 Export PDE Coefficient 选项,将参数 c 输出到 MATLAB 命令窗口中去。最后,进入 Solve 菜单并选择 Export Solution 选项,输出结果数组。

在需要的变量输出到 MATLAB 命令窗口后,用如下两个表达式来确定在 x 方向上的最大应力。

```
StressX = pdesmech(p,t,c,u,'tensor','sxx');
MaxSxx = max(StressX)
```

给出 Max Sxx = 301.469 9(N/cm²)

为了估计应力集中系数,将一个截面积为 $0.2d$ 的无凹槽平板应力 σ_{xx} 与上面求出的最大应力比较,无凹槽平板的应力 σ_{xx} 近似等于:

$$\sigma_{xx} = \frac{(100)(0.2)D}{0.2d} = 200\ \text{N/cm}^2$$

其中 $D/d = 2$,应力集中系数 S_{cc} 近似等于:

$$S_{cc} = \frac{301.4699}{200} \approx 1.50$$

这个值无论与实验获得的值 1.37 相比还是与纽伯(Nueber)图表算出的结果 1.45 相比都令人满意。①

注意:在使用 pdetool 来解决其他问题之前,应该使用 clear 来清除变量 p,t,c,a,变量使用新名就不必使用 clear。

练习

8.1 将图 8.39 中 $\boldsymbol{F}_1,\boldsymbol{F}_2,\boldsymbol{F}_3$ 及其合力表示为向量。确定结果值和以度数表示的方向余弦角。

[答案:$\boldsymbol{F}_1$ = [56.568 5, −42.426 4, −70.710 7]; $\boldsymbol{F}_2$ = [−26.948 9, −33.686 1, −67.372 2]; $\boldsymbol{F}_3$ = [28.640 1, 66.826 8, −95.466 9]; $\boldsymbol{R} = \boldsymbol{F}_1 + \boldsymbol{F}_2 + \boldsymbol{F}_3$ = [58.259 7, −9.285 7, −233.549 7], $|\boldsymbol{R}|$ = 240.885 6, α_R = [76.003 9, 92.209 2, 165.823 5]。]

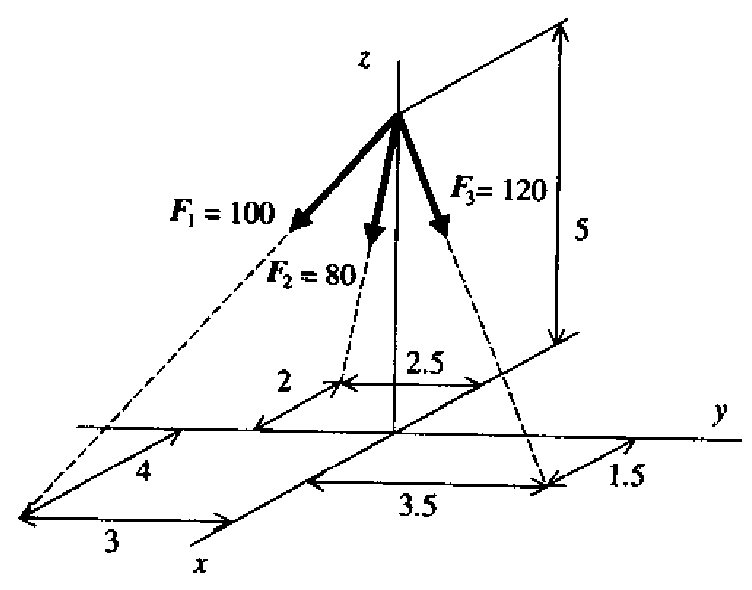


图 8.39 练习 8.1 各力的方向

① 参见 A.P.Boresi, R.J.Schmidt 和 O.M.Sidebottom, *Advanced Mechanics of Materials*, 第五版, John Wiley & Sons, New York, 1993 年, 第 582~584 页。

8.2 绳系固定在图 8.40 所示的坐标中。求力 F_1、F_2 和 F_3 的大小以及各自分量的大小。

[答案：F_1 = [−44.908 0,44.908 0,67.362 0]；| F_1 | = 92.5802；F_2 = [81.288 3 54.199 2,108.384 4]；| F_2 | = 145.917 0；F_3 = [0 −74.846 7,0]；| F_3 | = 74.846 7]。

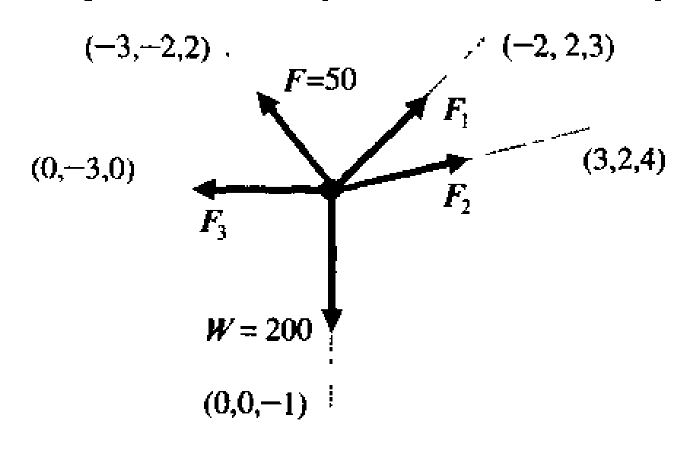


图 8.40　练习 8.2 各力的方向

8.3 用于振动的平面盘形从动件如图 8.41 所示，在(x, y)坐标系中，凸轮轮廓的坐标 R_x 和 R_y，刀具坐标 C_x 和 C_y 有如下的关系①，分别为：

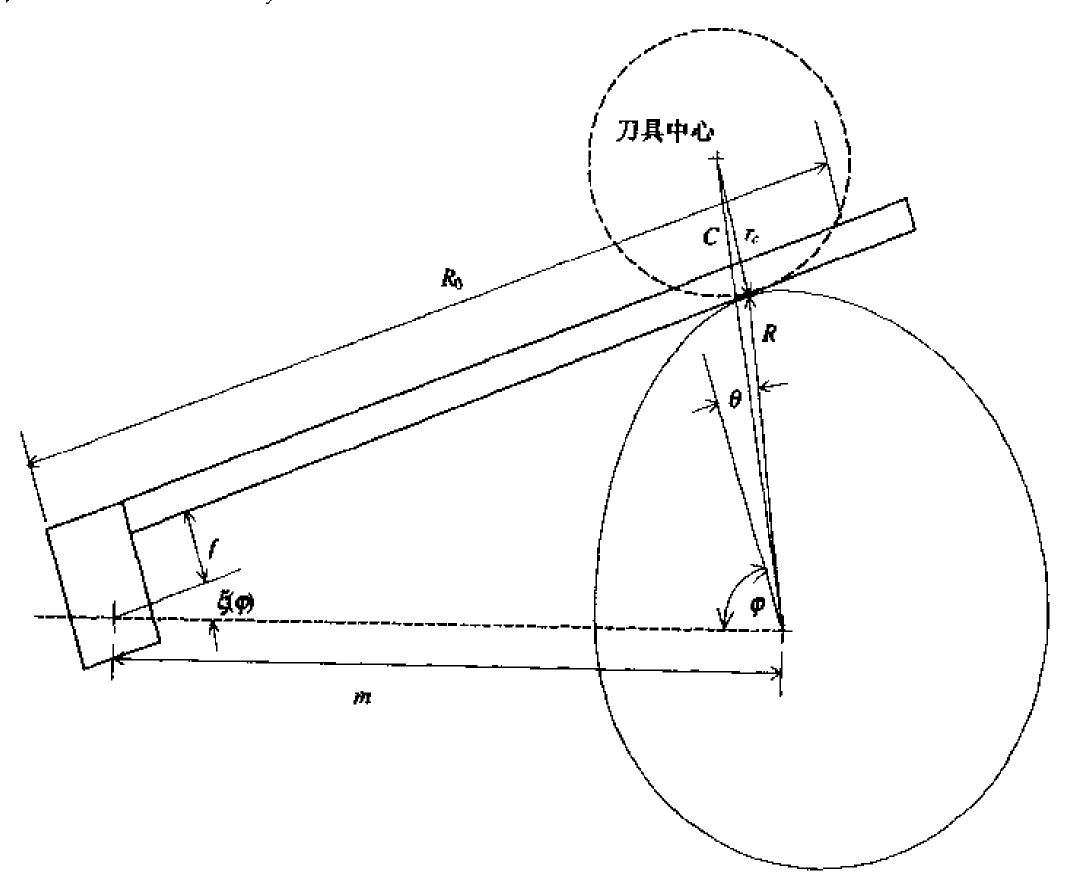


图 8.41　振动平面从动件

① 参见 A G Erdman 和 G.N.Sandor，出处同上，第 387 ~ 389 页。

$$R_x = R\cos(\theta + \varphi + \zeta_0) \qquad R_y = R\sin(\theta + \varphi + \zeta_0)$$
$$C_x = C\cos(\theta + \varphi + \zeta_0 - \gamma_0) \quad C_y = C\sin(\theta + \varphi + \zeta_0 - \gamma_0)$$

其中，

$$R = \frac{f + m\sin(\zeta)}{\cos(\theta)} \qquad \theta = \arctan\left[\left(\frac{\mathrm{d}\zeta/\mathrm{d}\varphi}{1 - \mathrm{d}\zeta/\mathrm{d}\varphi}\right)\frac{m\cos(\zeta)}{f + m\sin(\zeta)}\right]$$

$$C = \sqrt{c_x^2 + c_y^2} \qquad c_x = r_c\sin(\theta) \qquad c_y = R + r_c\cos(\theta)$$

$$\gamma_0 = \arctan\left(\frac{c_y}{c_x}\right) \qquad \zeta_0 = \arcsin\left(\frac{r_b - f}{m}\right) \qquad \zeta = \zeta(\varphi) = \zeta_0 + s(\varphi)/R_0$$

对于点 R_o的摆线运动，$s(\varphi)$在第 8.5.1 节中给出。当 $\beta = 60°$，$r_b = 3.25$，$h = 0.5$，$r_c = 0.5$，$f = 0.5$，$m = 5$，$R_o = 9$ 时，画出一个与图 8.29 和图 8.31 相似的图形。

8.4 考虑如图 8.42 所示的一个贝尔维尔圆锥弹簧。无量纲载荷 P'，无量纲弹簧比(弹性常数)k'和无量纲最大压缩应力σ'，均作为弹性形变 δ 的函数①，每个量的表达式如下：

$$P' = \frac{P}{P_0} = C_1 C_2 \qquad P_0 = \frac{Et^4}{(1 - v^2)b^2}$$

$$k' = \frac{k}{k_0} = C_2[1 + 1.5d_t^2 - 3d_t h_t + h_t^2] \qquad k_0 = \frac{P_0}{t}$$

$$\sigma'_{max} = \frac{\sigma'_{max}}{\sigma_0} = -d_t[C_3(h_t - 0.5d_t) + C_4] \qquad \sigma_0 = \frac{P_0}{t^2}$$

其中，

$$C_1 = 0.5d_t^3 - 1.5h_t d_t^2 + (1 + h_t^2)d_t$$
$$C_2 = \pi\left(\frac{\alpha + 1}{\alpha - 1} - \frac{2}{\ln(\alpha)}\right)\left(\frac{\alpha}{\alpha - 1}\right)^2$$
$$C_3 = \frac{\alpha^2}{(\alpha - 1)^2}\left(\frac{\alpha - 1}{\ln(\alpha)} - 1\right)$$
$$C_4 = \frac{\alpha^2}{2(\alpha - 1)}$$

并且 v 是泊松比，E 是弹性模量，$\alpha = b/a$，$h_t = h/t$，$d_t = \delta/t$。最大压缩应力发生在圆锥的上表面内边界 a 处并且通常规定$\sigma_{max} = \sigma_p/F_s$，其中 σ_p 是最大允许应力，F_s 是安全系数。

当 $d_t = 0.666\,7h_t$，$P' = 2.0$，$\sigma'_{max} = -4.5$ 时，求 α 和h_t 的值，以及 k'的值。

[答案：$\alpha = 2.269\,3$，$h_t = 1.484\,6$，$k' = 0.359\,8$]

---

① 参见 A H Burr 和 J B Cheatham, *Mechanical Analysis and Design*, 2nd ed Rrentice Hall, Upper Sanddle River, NJ, 1995 年，第 652 ~ 656 页。

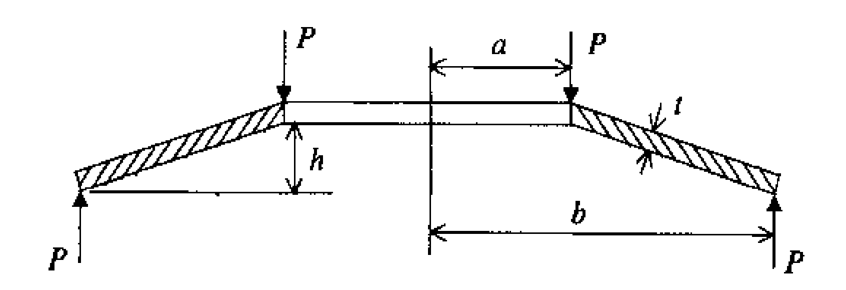


图 8.42　比立维尔圆锥弹簧的横截面

8.5　参考例 8.20 和图 8.22,确定 θ_2 的值,使 θ_4 取最大或最小值。[提示:θ_2 有两个值,一个对应 θ_4 最小值,另一个对应 θ_4 最大值,使用两个查找范围:π/2 到 π 和 3π/2 到 2π。在第二个查找范围内,θ_4 的符号应改变。]

[答案:$\theta_2 = 153.434\,5°$,$\theta_4 = 195.563\,9°$,$\theta_2 = 319.848\,9°$,$\theta_4 = 244.532\,9°$]

8.6　参见图 8.43,承受周期载荷的螺旋压缩弹簧的尺寸由以下方程决定。第一个是安全系数 F_s,取决于弹簧圈中的应力。

$$F_s = \frac{S_{eL}(S_u - \sigma_L)}{S_{eL}(\sigma_m - \sigma_L) + S_u\sigma_a} > 1$$

其中,

$$\sigma_L = K_s F_{min} K_1 \qquad K_1 = \frac{8C}{\pi d^2} \qquad C = \frac{D}{d}$$

$$\sigma_m = K_s F_m K_1 \qquad F_m = 0.5(F_{max} + F_{min})$$

$$\sigma_a = K_w F_a K_1 \qquad F_a = 0.5(F_{max} - F_{min})$$

$$K_s = 1 + \frac{1}{2C} \qquad K_w = \frac{4C-1}{4C-4} + \frac{0.615}{C}$$

$$S_u = 0.67Ad^b \qquad S_e = \frac{0.707 S_{el} S_u}{S_u - 0.707 S_{eL}}$$

F_{max}和 F_{min}分别是作用在弹簧上的最大和最小载荷。S_{el}是无限寿命疲劳极限,S_e是反向载荷的疲劳极限,S_u是极限剪切强度,D 是弹簧的平均直径,d 是弹簧钢线的直径。对于冷拔钢丝,常量 $A = 141\,040$ psi,常量 $b = -0.182\,2$。对于硬化弹簧钢,$d <$ 0.4 in, $S_{eL} = 67\,500$ psi。

第二个方程确定弹簧比(常数)k,由下式给出:

$$k = \frac{dG}{8N_a C^3}$$

式中,G 是这种材料的剪切弹性模量,N_a 是弹簧的匝数。对于钢 $G = 11.5 \times 10^6$psi。

最后的方程确定弹簧的自然长度 L_f,弹簧圈的直径 d,使得钢弹簧不弯曲的最大负荷 F_{max}。两端连接时,有下式:

$$\frac{F_{max}}{kL_f} = 0.812\,5\left[1 - \sqrt{1 - 6.865(Cd/L_f)^2}\right]$$

式中,$Cd = D$,并且:

$$L_f = d(Na + 2) + 1.15F_{max}/k - 0.15F_{min}/k$$

提供 15%的冲撞允许范围。

对于 $N_a=7$, $F_{max}=200$ lb, $F_{min}=40$ lb, $F_s=1.4$ 和 $k=125$ lb/inch 的硬化弹簧钢,求出 D 和 d,问弹簧是否会发生弯曲?

[答案: $d=0.251$ in, $D=1.868\,2$ in, $C=7.443\,2$。弹簧没有发生变形。]

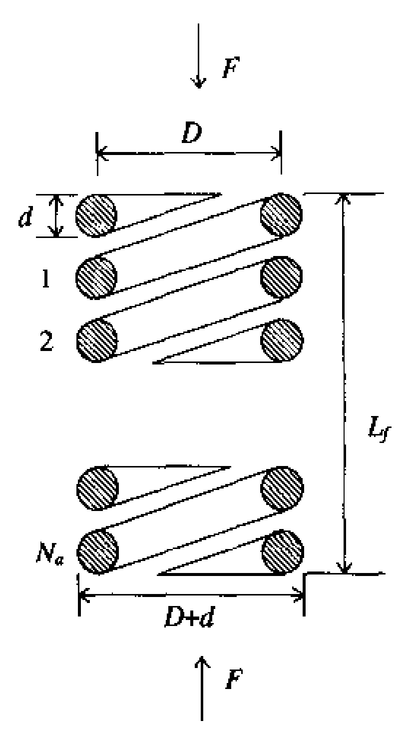


图 8.43　关于螺旋弹簧的术语

8.7 用例 8.21 中的结果绘出连杆的三个位置点,仿照图 8.26 标出点 P_1 的路径。用 fsolve 通过下式计算 Ψ 和 γ 值,从而确定 P_1 的路径,对 $\phi(\theta\leqslant\phi\leqslant\theta+\phi_2)$16 等分:

$$\mathbf{Z}_2(e^{j\phi}-1)-\mathbf{Z}_4(e^{j\Psi}-1)+(\mathbf{Z}_5-\mathbf{Z}_6)(e^{j\gamma}-1)=0$$

设上式中的实部和虚部均为 0,可得到 fsolve 所需的两个方程,对方程不使用代数的方法而使用数值计算的方法。

8.8 驱动螺杆升起载荷 W 所需的转矩为:

$$T_{raise}=\frac{Wd_p}{2}\left[\frac{\mu+\tan(\lambda)\cos(\alpha)}{\cos(\alpha)-\mu\tan(\lambda)}+\mu_c\frac{d_c}{d_p}\right]$$

式中 λ 是超前角,由下式求出:

$$\tan(\lambda)=\frac{L}{\pi d_p}$$

μ 是螺纹的摩擦系数,μ_c 是环管(螺母)的摩擦系数,d_p 是螺纹的节距直径,d_c 是螺母的平均半径,$L=mp$ 是螺距,$p=1/N_t$ 是节距,N_t 是每英寸螺纹的头数,m 是起始螺纹的头数。对于 Acme 螺纹来说,角度 α 为 14.5°,$1.9°<\lambda<6°$,螺纹和螺母的效率 e 为:

$$e=\frac{WL}{2\pi T_{raise}}=\tan(\lambda)\left[\frac{\mu+\tan(\lambda)\cos(\alpha)}{\cos(\alpha)-\mu\tan(\lambda)}+\mu_c\frac{d_c}{d_p}\right]^{-1}$$

当 $\mu_c=0$ 时(见练习 6.3),e 由下式表示:

$$e=\frac{\cos(\alpha)-\mu\tan(\lambda)}{\cos(\alpha)-\mu\cot(\lambda)}$$

以每分钟 v_r 英尺的速度升起载荷需要的马力为：

$$hp = \frac{v_r T_{raise}}{262\,6 d_p}$$

式中 T_{raise} 的单位是 in-lb。

对于截距直径为 1.0 in，超前角为 4°的一条单 Acme 线（$m=1$），当环管的平均直径为 1.25 in 时，求以每分钟 15 ft 的速率升高 800 lb 所需的转矩和马力。线的摩擦系数为 0.13，环管的摩擦系数为 0.04。问这个系统的效率是多少？

[答案：$T_{raise}=99.858\,3$ in-lb，$e=25.501\%$，$hp=0.570\,4$。]

8.9 在一个扁节链中，连接板的疲劳强度决定功率极限为：

$$hp = K_s N_1^{1\,08} n_1^{0\,9} p^{3-0\,07/p}$$

式中，N_1 是较小链轮的齿数，n_1 是较小链轮的转速，单位是 r/m。p 是截距，也是链栓中心之间的距离，对于常规重量链条 $K_s=0.004$。当 $N_1=21$ 齿、$n_1=175\,0$ r/m 并且 $hp=10$ 时，求节距。

[答案：$p=0.4788\,1$。]

8.10 对于一个承受垂直的静（不变的）传输载荷 F_t 的齿轮，接触应力为：[1]

$$\sigma_c = Z_E\sqrt{\frac{K_v K_H F_t}{d_w b Z_I}} \quad \text{N/mm}^2$$

式中，Z_I 是抗点蚀的几何系数，小齿轮（齿轮 1）的节距直径 $d_w=2R_{p1}$，R_{p1} 在表 8.4 中定义，Z_E 是弹性系数。其余的值在第 8.3 节中定义。

弹性系数的公式如下：

$$\frac{1}{Z_E} = \sqrt{\pi\left[\frac{1-v_1^2}{E_1}+\frac{1-v_2^2}{E_2}\right]}$$

式中，v_1 和 v_2 分别是小齿轮和齿轮的泊松比，E_1 和 E_2 分别是小齿轮和齿轮的弹性模量，当小齿轮和齿轮都是钢材料时，$Z_E=190(\text{N/mm}^2)^{1/2}$。

抗点蚀几何系数为：

$$Z_1 = \frac{\rho_1\rho_2\cot(\varphi)}{Cd_p}$$

$$\rho_1 = \sqrt{R_{T_1}^2 - R_{b_1}^2} - m\pi\cos(\varphi_s)$$

$$\rho_2 = (R_{b_1}+R_{b_2})\tan(\varphi) - \rho_1$$

其中，小齿轮的节圆直径 $d_p=N_1$m，其余数值在表 8.4 和表 8.5 中定义。

要想计算出接触应力 σ_c，必须确定允许应力 σ_H，并验证 σ_H 大于 F_t 引起的接触应力，即 $\sigma_c \leqslant \sigma_H$。允许接触应力如下：

$$\sigma_H = \frac{\sigma_{HP} Z_N Z_w}{F_{Sc} Y_Z}$$

下式适用于油和齿轮的温度低于 120℃的情况。F_{Sc} 值是点蚀的安全系数，Y_z 是第

---

[1] 参见 J R Colbourne，出处同上，1987 年。

8.3 节中定义的可靠系数，Z_N是抗点蚀应力循环系数，σ_{HP}是对于点蚀允许的接触应力数，Z_w是抗点蚀的硬度比例系数。

完全硬钢齿轮的抗点蚀允许的接触应力值由下式估计：

$$\sigma_{HP} = 2.41B_H + 237$$

适用于二级钢。

下式：

$$\sigma_{HP} = 2.22B_H + 200$$

适用于一级钢，布里涅耳硬度范围是 $180 \leqslant B_H \leqslant 400$。

抗点蚀应力循环系数由下式给出：

$$Z_N = 1.4723 \qquad n_L < 10^4$$

$$Z_N = 2.466 n_L^{-0.056} \qquad n_L \geqslant 10^4$$

其中 n_L是单向载荷循环次数。

全硬齿轮驱动的小齿轮表面抗点蚀硬度比系数用下式估算：

$$Z_w = 1 + 0.00075 e^{-0.448R_z}(450 - B_{H_2}) \quad R_z \leqslant 1.6$$

$$Z_w = 1 \qquad R_z > 1.6$$

其中 R_z 是小齿轮的表面磨光度，单位是 μm；B_{H_2}是齿轮的布里涅耳硬度，范围是 $180 \leqslant B_{H_2} \leqslant 400$。

考虑下面一对钢齿轮：

$m = 10$ mm　　$b = 45$ mm　　$n = 1\,800$ r/m　　$B_{H_2} = 260$

$N_1 = 28$　　$\varphi_s = 20°$　　$T = 2\,500$ Nm

$N_2 = 75$　　$C = 525$ mm　　$R_{T_1} = 153.9$ mm

根据下列条件确定接触应力和允许接触应力：

安全系数是 1.2

单向载荷循环次数是 4×10^8

小齿轮的表面磨光度是 1.1 μm

失效率希望低于 1/100

$Q_v = 8$

该齿轮是表 8.3 中定义的二型齿轮。

[答案：$\sigma_c = 857.2436$ N/mm², $\sigma_H = 636.2982$ N/mm²。]

8.11 使用第 8.2.1 节中的技术，绘出梁的无量纲挠度、转角、弯矩和沿梁的剪力。根据下列边界条件和载荷，给出前述各量的极大值和位置。

(a)在 $\eta = 0$ 和 $\eta = 1$ 处固定，沿梁有均布载荷。[按此条件用解析法① 求出 $y_{max}(\eta = 0.5) = -1/384$，$\theta_{max}(\eta = 0.2113$ 或 $\eta = 0.7887) = -0.0080$，$M_{max}(\eta = 0$ 或 $1) = -1/12$，$V_{max}(\eta = 0$ 或 $1) = 0.500$。]

① 参见 W Beitz 和 K H Kuttner，出处同上，第 B24 页。

(b)在 $\eta=0$ 处简单支撑,在 $\eta=1$ 处固定。沿梁有均布载荷。[按此条件用解析法[1]得到 $y_{max}(\eta=0.4215)=-1/185, \theta_{max}(\eta=0)=-1/48, M_{max}(\eta=1)=-1/8, V_{max}(\eta=1)=5/8$。]

(c)在 $\eta=0$ 和 $\eta=1$ 处固定。在 $\eta=0$ 处为 1,在 $\eta=1$ 处为 1 时为三角形载荷。[按此条件用解析法[2]得到 $y_{max}(\eta=0.525)=-1/764, \theta_{max}(\eta=0.8077)=0.00427, M_{max}(\eta=1)=-1/20, V_{max}(\eta=1)=-7/20$。]

8.12 外部增压中心凹槽推进空气轴承如图 8.44 所示。使用表 8.7 的定义,给定 P_s,通过轴承的正常物质流由下式给出[3]:

$$m' = P_s\sqrt{\left(\frac{P_1}{P_s}\right)^{2/k} - \left(\frac{P_1}{P_s}\right)^{(k+1)/k}}$$

其中 $m'=m/m_o, P_1=p_1/p_a, P_s=p_s/p_a$,$P_1$的解为:

$$P_1 = \sqrt{1 + BP_s^{(k-1)/2k}P_1^{1/k}\sqrt{P_s^{(k-1)/k} - P_1^{(k-1)/k}}}$$

当 $P_1/P_s=p_1/p_s>p_c=[2/(k+1)]^{k/(k-1)}$时,$P_1$的解为:

$$P_1 = \sqrt{1 + BP_sP_c^{1/k}\sqrt{1 - P_c^{(k-1)/k}}}$$

当 $P_1/P_s \leqslant P_c$ 时,是数据流拥塞的区域[马赫(Mach)数等于 1]。

轴承通常能够承受的载荷为:

$$W' = \frac{\sqrt{\pi}}{2}\frac{r_1^2}{r_2^2}A_2\exp(P_1^2/A_2^2)\left[\mathrm{erf}\left(\frac{P_1}{A_2}\right) - \mathrm{erf}\left(\frac{1}{A_2}\right)\right]$$

其中 $W'=W/(\pi r_2 Pa)$,erf 是误差函数,并且:

$$A_2 = \sqrt{\frac{(1-P_1^2)}{2\ln(r_1-r_2)}}$$

(a)当 $B=2$ 时,分别确定(i)$P_s=2$ 和(ii)$P_s=10$ 时的 P_1 值。在使用函数 fzero 时,设置的搜索范围为[0.1 P_s]。

[答案:(i) $P_1=1.4016$,(ii)$P_1=2.4852$。]

(b)当 $r_1/r_2=0.1, B=2$ 时,对应(i)$P_s=2$ 和(ii)$P_s=10$ 时,确定 m'和 W'的值。

[答案:(i)$m'=0.4822, w'=0.0954$;(ii)$m'=2.5880, W'=0.4147$。]

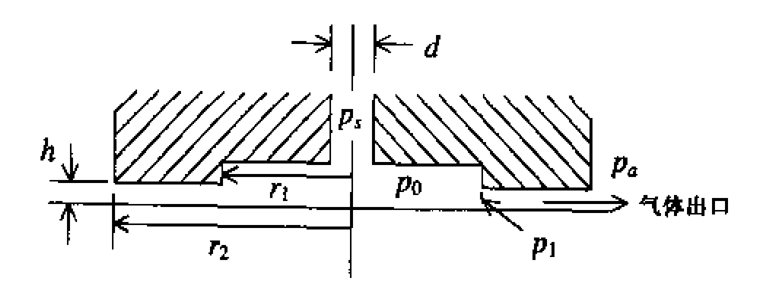


图 8.44 空气轴承的几何尺寸

① 参见 W Beitz 和 K H Kuttner,出处同上,第 B23 页。
② 参见 W Beitz 和 K H Kuttner,出处同上,第 B24 页。
③ 参见 W A Gross, *Gas film Lubrication*, John Wiley & Sons, New York, 1962 年,第 5 章。

表 8.7 空气轴承公式中量的定义

量	定义
C_D	泄露系数
$A_o = \pi d^2/4$	限制域
P_s	提供压力
P_o	凹口压力
p_1	入口压力
p_a	周围压力
h	气膜厚度
μ	气体粘度
k	热比(空气为 1 4)
R	气体常数(通用气体常数除以分子量)
θ	温度(绝对温度)
$B = \frac{12C_D\mu A_o}{h^3 p_a \pi}\ln\left(\frac{r_2}{r_1}\right)\sqrt{\frac{(k-1)R\theta}{2k}}$	轴承参数
$m_o = \frac{1}{C_D A_o p_a}\sqrt{\frac{(k-1)R\theta}{2k}}$	质量流参数
m	质量流
W	轴承载荷

8.13 使用第 8.7 节介绍的过程确定有圆形孔或椭圆形孔的平板上的最大应力。设有三种情况如图 8.45 所示。假定孔的边界无应力。对于这三种情况,令 $\sigma = 1$,表示 y 轴的正方向;$\sigma = -1$,表示 y 轴的负方向。再令弹性模量等于 200×10^3,使用一次 Refine Mesh 选项。对于情况(a),令 $a = 1$(单位);对于情况(b)和(c),令 $b = 1$(单位)和 $a = 2$。

[答案:

情况(a):对于圆形孔的情况,用解析法获得的最大应力比① 是 $\sigma_{yy}/\sigma = 3$;用 PDE 方法给出 2.912 2,或低于 2.9%。

情况(b):对于椭圆形孔且载荷垂直于椭圆长轴的情况,用解析法获得② 的结果是:$\sigma_{yy}/\sigma = 1 + 2a/b$,数值 $\sigma_{yy}/\sigma = 5$。用 PDE 方法给出的是 4.226 4,低于 15.5%。

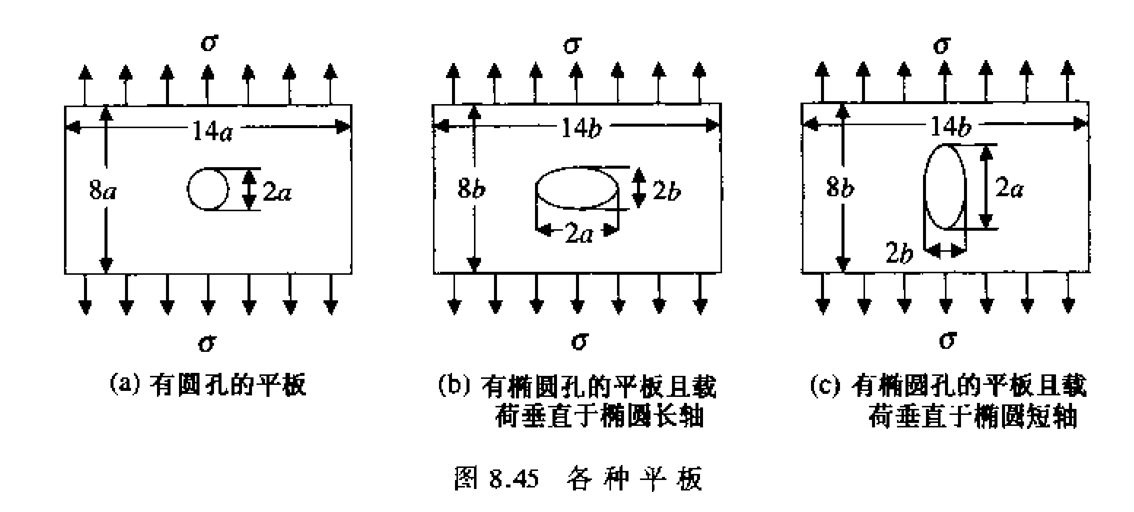
情况(c):对于椭圆形孔且载荷垂直于椭圆短轴的情况,用解析法获得③ 结果是:$\sigma_{yy}/\sigma = 1 + 2b/a$,数值 $\sigma_{yy}/\sigma = 2$。用 PDE 方法给出的是 2.355 2,高出 17.8%。]

---

① 参见 Boresi 等,出处同上,第 566 ~ 567,569 页。

② 参见 Boresi 等,出处同上,第 568 ~ 569 页。

③ 参见 Boresi 等,出处同上,第 570 页。



(a) 有圆孔的平板　(b) 有椭圆孔的平板且载荷垂直于椭圆长轴　(c) 有椭圆孔的平板且载荷垂直于椭圆短轴

图 8.45　各种平板

参考文献

1. Fundamental Rating Factors and Calculation Methods for Involute Spur and Helical Gear Teeth, AGMA Standard ANSI/AGMA 2002 - C95. American Gear Manufacturers Association, 1500 King Street, Alexandria VA, 22314.
2. Beitz W and Kuttner K H. Eds. Handbook of Mechanical Engineering. New York, Springer-Verlag, 1994.
3. Bhonsle S R and Weinmann L J. Mathematical Modeling for the Design of Machine Components. NJ: PrenticeHall, Upper Saddle River, 1999.
4. Burr A H and Cheatham J B. Mechanical Analysis and Design. 2nd ed. NJ: Prentice Hall, Upper Saddle River, 1995.
5. Colbourne J R, The Geometry of Involute Gears. New York: Springer-Verlag, 1987.
6. Dimarogonas A D, Computer Aided Machine Design. NJ: Prentice Hall, Upper Saddle River, 1989.
7. Erdman A G and Sandor G N. Mechanical Design: Analysis and Synthesis. 2nd ed. NJ: Prentice Hall, Upper Saddle River, 1991.
8. Hibbeler R C. Engineering Mechanics. 8th ed. NJ: Prentice Hall, Upper Saddle River, 1998.
9. Jensen P J. Cam Design and Manufacture. 2nd ed. New York: Marcel Dekker, Inc., 1987.
10. Norton R L. Design of Machinery. New York: McGraw-Hill, Inc., 1992.
11. Norton R L. Machine Design, An Integrated Approach. NJ: Prentice Hall, Upper Saddle River, 1996.
12. Rothbart H A. Cams: Design, Dynamics and Accuracy. New York: John Wiley & Sons, 1956.
13. Shigley J E and Mischke C R. Mechanical Engineering Design. 5th ed. New York: McGraw-Hill, 1989.
14. Slocum A H. Precision Machine Design. NJ: Prentice Hall, Upper Saddle River, 1992.
15. Spotts M F and Shoup T E. Design of Machine Elements. NJ: Prentice Hall, Upper Saddle River, 1998。
16. Wilson C E. Computer Integrated Machine Design. NJ: Prentice Hall, Upper Saddle River, 1997.

第9章　动力学与振动

本章将介绍几种不同的方法，分别用于分析刚体动力学、单自由度和多自由度线性与非线性系统中的自由振动和强迫振动以及梁的振动。

9.1　轨迹

如图9.1所示，重力场中有两个物体，其中质量为 m_2 的物体固定，而质量为 m_1 的物体绕 m_2 做平面圆周运动。做圆周运动的 m_1 物体的轨道半径用变量 r 表示，角度用变量 θ 表示。

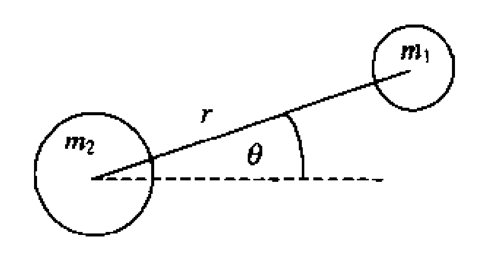


图9.1　两物体系统

卫星绕地球转动时，m_2 等于地球的质量，m_1 等于卫星的质量，r 为卫星球心与地球球心间的距离。其运动轨迹由下列方程组决定①②：

$$\frac{d^2 r}{d\tau^2} - r\left(\frac{d\theta}{d\tau}\right)^2 = -\frac{4\pi^2}{r^2} \tag{9.1}$$

$$r\frac{d^2\theta}{d\tau^2} + 2\frac{dr}{d\tau}\frac{d\theta}{d\tau} = 0$$

式(9.1)中，$\tau = t/P_c$，其中 t 是时间变量，P_c 为物体在地球表面做圆周运动的周期。在地球表面，r 约等于地球半径，即 $r = 6.373 \times 10^6$ m。

尽管由式(9.1)可得出非线性系统的一个封闭解③，但仍需求出微分方程在给定初始条件下的数值解。初始条件为：

$$r(0),\quad dr(0)/d\tau, \theta(0),\quad d\theta(0)/d\tau$$

有时可简写为：

$$(r(0), dr(0)/d\tau, \theta(0), d\theta(0)/d\tau)$$

式(9.1)给出了动态微分系统的一个例子，描述了状态变量 r 和 θ 随 τ 的变化。一般来说动态微分系统由一组微分方程构成，描述所研究的状态随独立变量（比如时间）的变化而变化的情况，正如式(9.1)所示，当一个动态系统由一组常微分方程构成时，可以用对独立变量的数

---

① 参见 D T. Greenwood, *Principles of Dynamics*，第二版，Prentice Hall，Englewood Cliffs, NJ, 1998年，第5章。

② 参见 F.C. Moon, *Applied Dynamics with Applications to Multibody and Mechatronic Systems*, John Wiley & Sons, New York, 1998年，第7章。

③ 参见 D.T. Greenwood，出处同上。

值积分来求出该公式在给定初始条件下的解。至此，把式(9.1)化为下列一阶微分方程组，选用 ode45 函数来实现求解。引入新状态变量：

$$x_1 = r, \qquad x_3 = \theta$$

$$x_2 = \frac{\mathrm{d}r}{\mathrm{d}\tau}, \qquad x_4 = \frac{\mathrm{d}\theta}{\mathrm{d}\tau}$$

代人式(9.1)可得四个一阶微分方程：

$$\begin{aligned} \frac{\mathrm{d}x_1}{\mathrm{d}\tau} &= x_2 \\ \frac{\mathrm{d}x_2}{\mathrm{d}\tau} &= x_1 x_4^2 - \frac{4\pi^2}{x_1^2} \\ \frac{\mathrm{d}x_3}{\mathrm{d}\tau} &= x_4 \\ \frac{\mathrm{d}x_3}{\mathrm{d}\tau} &= -\frac{2x_2 x_4}{x_1} \end{aligned} \tag{9.2}$$

用函数 orbit 表示式(9.2)：

```
function xdot = orbit(t,x)
xdot = [x(2);x(1) * x(4)^2 - 4.0 * pi^2/x(1)^2;x(4); - 2.0 * x(2) * x(4)/x(1)];
```

用 ode45 函数对 orbit 定义的微分方程组求解，针对表 9.1 所列出的三组初始条件求出物体 m_1 运动的三种轨迹。下面的程序代码求出了运动轨迹并画出了轨迹图形：

```
initcond = [2.0 0.0 0.0 1.5;1.0 0.0 0.0 2.0 * pi;2.0 0.0 0.0 4.0];
tspan = linspace(0,5,1000);
options = odeset('RelTol',1e - 6,'AbsTol',[1e - 6 1e - 6 1e - 6 1e - 6]);
lintype = ['-'.'-'.'-.'];
for i = 1:3
  [t,x] = ode45('orbit',tspan,[initcond(i,:)]',options);
  polar(x:,3),x(:,1),lintype(2^(i - 1) + 1:2^i));
  hold on
end
text(0.50, - 1 20,'Elliptical orbit');
text( - 1.20,1.00,'Circular orbit');
text(1.75,2.00,'Hyperbolic orbit');
```

数组 *initcond* 中的元素为三组初始条件的值，linspace 函数定义了确定时间内可用 1 000 个等间隔恒定值以保证数值分析过程中尽量不产生无效解。另外，odeset 函数设置相对误差精度等于 10^{-6}，还设置了四种状态的绝对误差精度——用 x_j 表示，$j = 1,2,\cdots 4$。

表 9.1　3 组初始条件

组	$x_1(0)$	$x_2(0)$	$x_3(0)$	$x_4(0)$	轨迹类型
1	2.0	0.0	0.0	1.5	椭圆
2	1.0	0.0	0.0	2π	圆
3	2.0	0.0	0.0	4.0	双曲线

图 9.2 是上述程序执行的结果。由第一种初始条件得到的轨迹为椭圆，由第二种初始条件得到的轨迹为圆，第三种初始条件得到的轨迹为双曲线，即一非闭合轨迹，表示物体的运动为无限运动①，在这种情况下卫星已脱离绕地球转动的轨道，而不再是一般的卫星运动了。

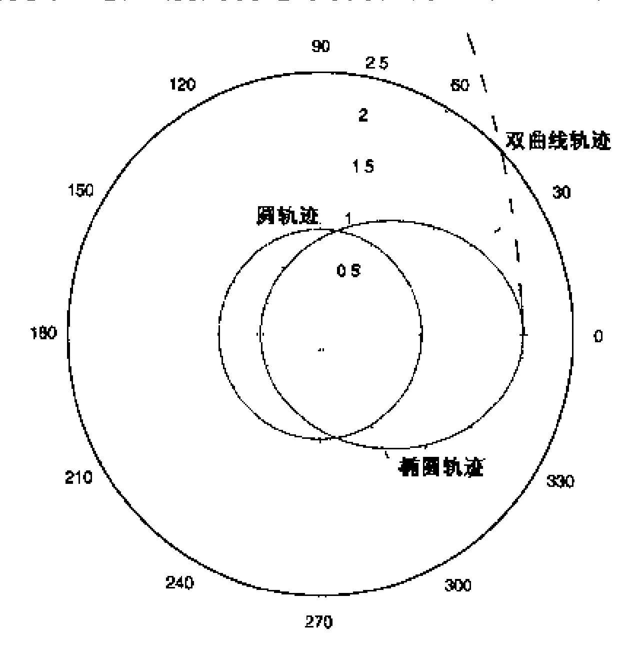


图 9.2　根据表 9.1 给定的初始条件由式(9.1)推出的三种轨迹

用数值积分方法分析类似式(9.1)描述的系统，积分过程中可能会产生无效解。一般系统中有一个状态量不随时间变化[例如式(9.1)中单位质量的角动量]，当产生无效解时，该常量也将随时间变化。无阻尼系统进行数值分析时可根据系统的动能和势能之和是否保持不变来判断是否产生无效解。因此较好的方法是通过改变步长或改变误差精度，或两者同时改变来确定结果是否不同(参考练习 9.1)。

9.2 单自由度系统

9.2.1 概述

如图 9.3 所示，在弹簧—质量—阻尼器系统中，振体质量为 m，弹簧的线性系数(倔强系数)为 k，非线性系数为 α，阻尼器的阻尼系数为 c。振体受一外力 $F(t)=X_0kf(t)$作用。该方程是研究力学系统的一种典型模型，研究范围可以包括从洗衣机到车辆的振动问题。②

若用 x 表示系统的位移，则运动方程具有下列形式：

$$\frac{d^2x}{d\tau^2}+2\zeta\frac{dx}{d\tau}+x+\hat{\alpha}x^3=X_of(\tau) \tag{9.3}$$

① 参见 D T Greenwood，出处同上，第 211 页。

② 参见 D J. Inman, *Engineering Vibration*, Prentice Hall, Englewood Cliffs, NJ, 1994; S S Rao, *Mechanical Vibrations*，第三版，Addison - Wesley, Reading, MA, 1995 年; B H. Tongue, *Principles of Vibration*, Oxford University Press, New York, 1996 年。

式中 $\tau=\omega_n t$ 为无量纲时间，阻尼因子 ζ 由下式求出：

$$\zeta=\frac{c}{2m\omega_n} \tag{9.4}$$

式(9.4)中的固有频率 ω_n 为：

$$\omega_n=\sqrt{k/m} \tag{9.5}$$

非线性系数 $\hat{\alpha}$ 由下式确定：

$$\hat{\alpha}=\frac{\alpha}{m\omega_n^2} \tag{9.6}$$

在式(9.3)给出的非线性系统中，当 $\hat{\alpha}$ 为负值时弹簧为软弹簧，当 $\hat{\alpha}$ 为正值时弹簧为硬弹簧①。

引入新变量或状态量：

$$x_1=x$$

$$x_2=\frac{dx}{d\tau}$$

式(9.3)转化为状态空间方程形式：

$$\begin{aligned}\frac{dx_1}{d\tau}&=x_2\\ \frac{dx_2}{d\tau}&=-2\zeta x_2-x_1-\hat{\alpha}x_1^3+X_o f(\tau)\end{aligned} \tag{9.7}$$

在分析式(9.7)的几种解之前，先创建函数 FreeOcillation，以被函数 ode45 调用：

```
function xdot = FreeOscillation(t, x, dummy, zeta, AlphaHat)
xdot = [x(2); -2 * zeta * x(2) - x(1) - AlphaHat * x(1)^3];
```

对于线性系统，有 $AlphaHat=\hat{\alpha}=0$。

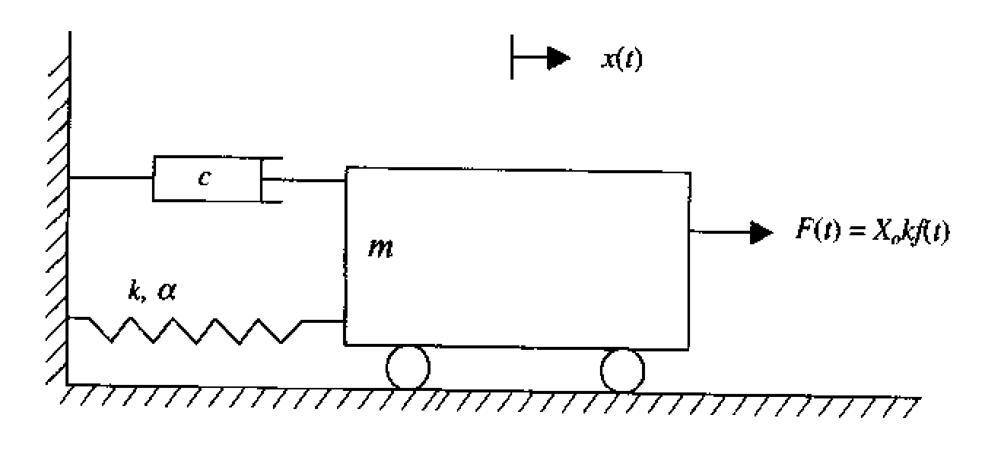


图 9.3　弹簧—质量—阻尼器系统

9.2.2　线性系统的自由振动

当 $\hat{\alpha}=F(\tau)=0$ 时，得到一线性系统的自由振动方程，式(9.3)变为：

---

① 参见 A H. Nayfeh 和 B. Balachandran, *Applied Nonlinear Dynamics: Analytical, Computational, and Experimental Methods*, John Wiley & Sons, New York, 1995 年。

$$\frac{d^2x}{d\tau^2} + 2\zeta\frac{dx}{d\tau} + x = 0 \tag{9.8}$$

其特征方程为：

$$\lambda^2 + 2\zeta\lambda + 1 = 0 \tag{9.9}$$

当 $\zeta = 0.1$ 时，值可由：

```
lambda = roots([1 0 2 1])
```

求得，结果为：$\lambda_1 = -0.1000 + 0.9950i$ 和 $\lambda_2 = -0.1000 - 0.9950i$，$i = \sqrt{-1}$。两者的实部均为负值，表明系统是稳定的，这是欠阻尼系统的特征[①]。在欠阻尼系统中有：

$$|\mathrm{Re}(\lambda_j)| = \zeta \quad 和 \quad |\lambda_1| = |\lambda_2| = 1$$

下面求式(9.8)表示的线性弹簧—质量—阻尼器系统的自由振动响应。给定初始条件：

$$x(0) = x_o$$

$$\frac{dx(0)}{d\tau} = v_o$$

以及三种阻尼系数 ζ：(1)$\zeta = 0.1$，为欠阻尼情况；(2)$\zeta = 1$，为临界阻尼情况；(3)$\zeta = 5.0$，为过阻尼情况。系统在初始条件 $x_o = 1$ 和 $v_o = 1$ 下开始振动。

下面的程序用于求 $0 \leqslant \tau \leqslant 40$ 范围内的自由振动响应：

```
zeta = [0.1 1.0 5.0]; AlphaHat = [0.0 0.0 0.0];
tspan = linspace(0,40,400);
lintyp = ['-' '--' '-'];
for i = 1:3
 [t,x] = ode45('FreeOscillation',tspan,[1 1]',[],zeta(i),AlpaHat(i));
 subplot(2,1,1);
 plot(t,x(:,1),lintyp(2 * (i - 1) + 1:2 * i));
 hold on
 subplot(2,1,2);
plot(x(:,1),x(:,2),lintyp(2 * (i - 1) + 1:2 * i));
hold on
end
subplot(2,1,1)
xlabel('Time ( \ tau)');
ylabel('Displacement x( \ tau)');
title('Displacement as a function of \ tau');
axis([0 40 - 2.0 2.0]);
text(2.7, - 1.3,' \ zeta = 0.1');
text(3.6, - 0.1,'1.0');
text(3.6,1.0,'5.0');
subplot(2,1,2)
xlabel('Displacement');
ylabel('Velocity');
title('Phase portrait')
axis([ - 2.0 2.0  - 2.0 2.0]);
text(0.7, - 1.25,' \ zeta = 0.1');
```

① 参见 D J Inman，出处同上，第 19 页；S S Rao，出处同上，第 130 页；B.H Tongue，出处同上，第 25 页。

```
text(0.8,-0.65,'1.0');
text(0 8,0 1,'5 0');
```

程序的执行结果如图9.4所示。上图为位移响应,下图为距离对速度的曲线图。点$(x,\dot{x})$由位移和速度坐标值组成,称为相点。不同初始条件下得到的相轨迹可组成一幅相平面图。式(9.8)确定的线性系统的平衡点对应于相平面图中 $x_1=0$ 和 $x_2=0$ 的点[或点(0.0,0.0)]。由位移响应图上看出只有欠阻尼情况有振动,对应到相平面图上是螺旋形曲线轨迹,随着时间的增加它趋向于点(0.0,0.0),该点为平衡点,也是一吸引点。在临界阻尼和过阻尼情况下,在系统到达平衡位置过程中并没有发生振动。对临界阻尼系统,到达平衡位置的时间最短。

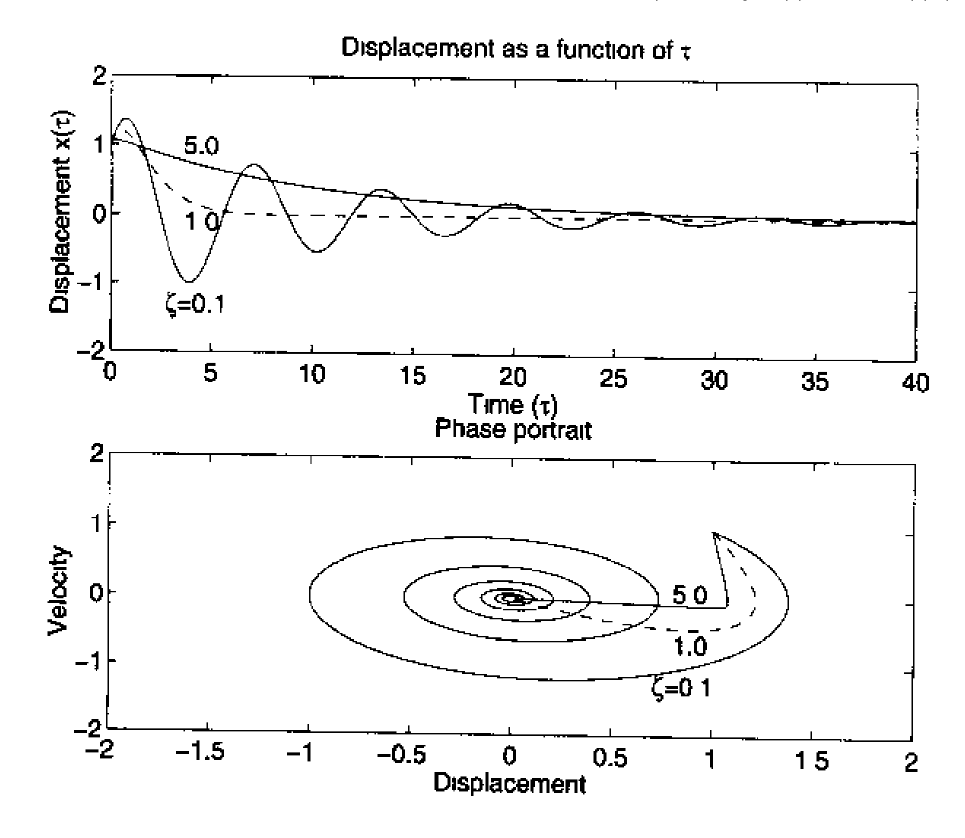


图9.4 阻尼线性系统自由振动的位移响应图及相平面图

现由欠阻尼系统的位移响应曲线估计阻尼因子,首先确定对数衰减率 δ,δ 由下式求得[①]:

$$\delta=\frac{1}{n}\ln\left(\frac{x_j}{x_{j+n}}\right) \tag{9.10}$$

阻尼因子与 δ 有关,具体定义如下:

$$\zeta=\frac{\delta^2}{\sqrt{4\pi^2+\delta^2}} \tag{9.11}$$

x_j 和 x_{j+n} 分别为时间 t_j 和 t_j+nT 对应的位移,T 为欠阻尼振动的周期。求 δ 的一种方法为先求不同的极大值(极小值)对应的幅度,然后从产生最大值处开始计算其周期 T。由这些结果可确定 x_j。图9.5为一欠阻尼系统的自由振动响应,参照此图,利用下面的函数对系统的阻尼系数进行了估计:

① 参见 D J,Inman,出处同上,第44页;S.S Rao,出处同上,第136页。

```
function[pspace,tref] = period(tdata,xdata,n,nskip)
[xmin0,tmin0] = min(xdata((1 + nskip):length(xdata)));
[xmax0,tmax0] = max(xdata((1 + nskip):length(xdata)));
tmin0 = tmin0 + nskip;tmax0 = tmax0 + nskip;
m = fix(length(tdata)/(abs(tmin0 - tmax0) * 2.0))
while n > m
 n = n - 1;
end
pspace = zeros(n,1);
tref = zeros(n + 1,1);
for j = 1:n
 if tmin0 < tmax0
  tref(1) = tmin0;
  if j = = 1
   [xmin 1,tmin 1] = min(xdata(tmax0:length(xdata)));
   pspace(j) = (tmax0 - tmin0) + tmin1;
   tref(j + 1) = tref(j) + pspace(j);
  else
   [xmaxj,tmaxj] = max(xdata(tref(j):length(xdata)));
   [xminj,tminj] = min(xdata(tref(j) + tmaxj:length(xdata)));
   pspace(j) = tmaxj + tminj;
   tref(j + 1) = tref(j) + pspace(j);
  end
 else
  tref(1) = tmax0;
  if j = = 1
   [xmax1,tmax1] = max(xdata(tmin0:length(xdata)));
   pspace(j) = (tmin0 - tmax0) + tmax1;
   tref(j + 1) = tref(j) + pspace(j);
  else
   [xminj,tminj] = min(xdata(tref(j):length(xdata)));
   [xmaxj,tmaxj] = max(xdata(tref(j) + tminj:length(xdata)));
   pspace(j) = tmaxj + tminj;
   tref(j + 1) = tref(j) + pspace(j);
  end
 end
end
```

period 函数的输入量为时间参数 *tdata*，对应的响应参数 *xdata*，循环次数 *nskip* 以及省略的初始数据点。period 函数的输出量包括数组 *pspace* 和 *tref*，它们分别为波峰（或波谷）间的间隔指标和与之对应的时间指标。函数 max 和 min 用于求波峰（极大值）对应的时间和波谷（极小值）对应的时间以及波峰与波谷之间的时间差。第二行和第三行分别求出第一个波谷和第一个波峰的位置。由于没有用 gradient 函数来检查极值是否存在，所以在确定极值点时可能会产生错误。例如，图 9.5 中显示的数据，如果 min 函数的起始时间设为 $z = 0$，则图中第一个数据点被误认为是第一个极小值。为避免此类错误，提供了一个输入参数 *nskip* 以定义数据的起始点。

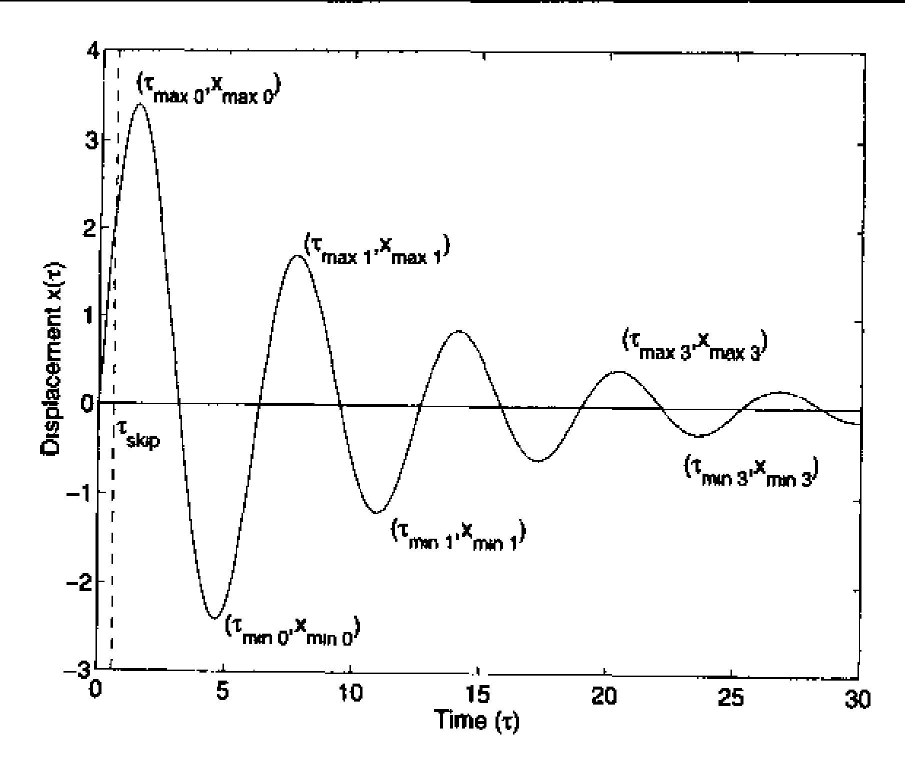


图 9.5　欠阻尼系统的自由振动响应

极大值(极小值)与其后继极小值(极大值)之间的间隔对应于阻尼振动的半个周期。在第五行利用这一点估算 *xdata* 中的总循环次数 m。如果定义的输入量 n 大于 m,则 n 被设置为小于或等于 m。从第九行开始对数组 *pspace* 和 *tref* 进行初始化,随后在 for 循环内求出极值(极大值或极小值)间的间隔。在循环体内还用到逻辑判断语句以区分第一个极值为极小值的情况和第一个极值为极大值的情况。后一种情况求出波峰之间的间隔指标以及相关的时间指标,对其他情况求出波谷之间的间隔指标以及相关的时间指标。由 period 函数求得的极值可以作为确定经过不同循环后阻尼振动的周期的依据。

尽管线性系统的阻尼振动周期在经过不同循环后不会改变,但对于非线性系统却并非如此。接下来的程序中利用 period 函数的输出量和式(9.10)估算式(9.8)的系统响应的对数衰减率。系统参数为 $\zeta = 0.3$ 和 $\hat{\alpha} = 0.0$,响应的起始点为 $x_0 = 0.0$ 和 $v_0 = -10.0$。

```
zeta = 0.3; AlphaHat = 0.0;
tspan = linspace(0.0,40.0,400);
[t,x] = ode45('FreeOscillation',tspan,[0-10]',[],zeta,AlphaHat);
n = input('Enter number of cycles');
nskip = input('Enter number of initial points to skip');
[pspace,tref] = period(t(:,1),x(:,1),n,nskip);
zeta = zeros(length(pspace),1);
fprintf(1,'Cycle Number Damping Factor \n')
for j = 1:length(pspace)
 logdec = log(x(tref(j),1)/x(tref(j+1),1));
 zeta(j) = sqrt(logdec^2/(4.0*pi^2+logdec^2));
 fprintf(1,'  %3d         %6.4f\n',j,zeta(j))
end
```

程序前两行用于产生式(9.8)系统的均匀分布的数据,然后调用 period 函数求出阻尼因子。在第 10 行和第 11 行分别实现式(9.10)和式(9.11)。给定 $n = 4$, $nskip = 4$ 时,运行程序,

在 MATLAB 命令窗口中显示如下内容：

```
Enter number of cycles 4
Enter number of initial points to skip 4
Cycle Number Damping Factor
    1          0.3008
    2          0.3006
    3          0.2992
    4          0.3004
```

正如所期望的，对一通过粘性阻尼而消耗能量的线性系统来说，衰减是指数衰减，而且在阻尼振动的每次循环中对数衰减率均保持不变。

上面讨论的程序用以介绍一种估算阻尼因子的方法。对该程序可加入逻辑判断从而排除那些末尾时间对应的数据点，并判断出数据是否为欠阻尼系统。当阻尼因子接近 1 时上述方法将产生错误。下面的程序给出了另一种方法，其输入为图形输入。

```
zeta = 0.3; AlphaHat = 0.0;
tspan = linspace(0.0,40.0,400);
[t,x] = ode45('FreeOscillation',tspan,[0 -1.0]',[],zeta,AlphaHat);
plot(t(:,1),x(:,1));
disp(['Choose the location of the first peak(valley)']);
[tj xj] = ginput(1);
disp(['Choose the location of the next peak (valley)']);
[tjn xjn] = ginput(1);
n = input('Enter number of cycles between chosen peaks (valleys)');
logdec = log(xj/xjn)/n;
disp(['Estimated zeta = 'num2str(sqrt(logdec^2/(4.0 * pi^2 + logdec^2)))])
```

函数 ginput 用于确定(t,x)的坐标值，这些从图中选择，参见第 6.3.7 节。

9.2.3 非线性系统的自由振动

现在研究两个非线性系统的振动。第一个系统中含有一个三次非线性弹簧，即式(9.3)在 $F(\tau)=0$ 且 $\hat{\alpha}\neq 0$ 时的情况。第二个系统中有一非线性阻尼器。

非线性弹簧系统 对式(9.3)描述的系统运动分三种情况加以检验，三种情况如表 9.2 所示。第一种情况为一线性系统，而第二和第三种情况为带软弹簧的非线性系统。前两种情况下的初始条件相同，第三种情况则不同。

表 9.2 三种情况下的参数和初始条件

情况	系统类型	$\hat{\alpha}$	ζ	$x(0)=x_o$	$dx(0)/d\tau=v_o$
1	线性	0.00	0.20	-2.00	2.00
2	非线性	-0.25	0.20	-2.00	2.00
3	非线性	-0.25	0.20	-2.00	2.31

第 9.2.1 节创建的函数 FreeOscillation 可以描述式(9.3)系统。下面的程序针对上述三种情况产生了不同的响应输出，另外在估算阻尼因子的同时还求出响应曲线在不同循环中的周期。

```
zeta = 0.20; AlphaHat = [0.00 -0.25 -0.25];
xo = [-2.00 -2.00 -2.00]; vo = [2.00 2.00 2.31];
tspan = linspace(0.0,40.0,401);
lintyp = ['-''--''-'];
options = odeset('RelTol',1e-8,'AbsTol',[1e-8 1e-8]);
for i = 1:3
 [t,x] = ode45('FreeOscillation',tspan,[xo(i) vo(i)]',options,zeta,AlphaHat(i));
 subplot(2,1,1)
 plot(t,x(:,1),lintyp(2*(i-1)+1:2*i))
 hold on
 subplot(2,1,2)
 plot(x(:,1),x(:,2),lintyp(2*(i-1)+1:2*i))
 hold on
 fprintf(1,'\n\n Case %3d\n',i)
 [pspace,tref] = period(t,x(:,1),3,8);
 zetaest = zeros(length(pspace),1);
 fprintf(1,'\n Cycle No.    Period    Damping Factor\n')
  for j = 1:length(pspace)
   per(j) = t(tref(j+1)) - t(tref(j));
   logdec = log(x(tref(j),1)/x(tref(j+1),1));
   zetaest(j) = sqrt(logdec^2/(4.0*pi^2+logdec^2));
   fprintf(1,'  %3d       %8.3f       %6.4f\n',j,per(j),zetaest(j))
 end
end
subplot(2,1,1)
xlabel('Time{\tau}');
ylabel('Displacement x(\tau)');
title('Displacement histories');
axis([0.0 40.0 -3.0 3.0]);
text(4.0,-1.15,'Case 1:Linear system');
text(4.0,2.2,'Case 3:Nonlinear system');
subplot(2,1,2)
xlabel('Displacement');
ylabel('Velocity');
title('Phase portraits');
axis([-3.0 3.0 -3.0 3.0]);
text(0.85,-1.45,'Case 1');
text(0.85,0.0,'Case 2');
text(2.0,0.0,'Case 3');
```

第 5 行定义了积分过程中用到的误差值以确保数值误差不致于产生无效解。

严格来说，对数衰减率在估算非线性系统的阻尼因子时不是很有效。这里可以说明基于非线性系统的“大”振幅估算的阻尼因子会产生错误解，而基于“小”振幅估算的阻尼因子可得出合理解。另外，此结果也说明非线性系统的振动周期与响应振幅有关。该程序运行结果在 MATLAB 命令窗口上输出每种情况下经过不同循环时阻尼因子的估算值。

```
Case   1
Cycle No.       Period       Damping Factor
    1           6.500          0.2005
    2           6.400          0.1999
    3           6.400          0.1999
```

```
Case  2
Cycle No.        Period       Damping Factor
    1            7.300          0.2328
    2            6.400          0.2008
    3            6.400          0.2000
Case  3
Cycle No.        Period       Damping Factor
    1            9.000          0.2592
    2            6.500          0.2026
    3            6.400          0.1998
```

三种情况下的自由振动图如图 9.6 所示。

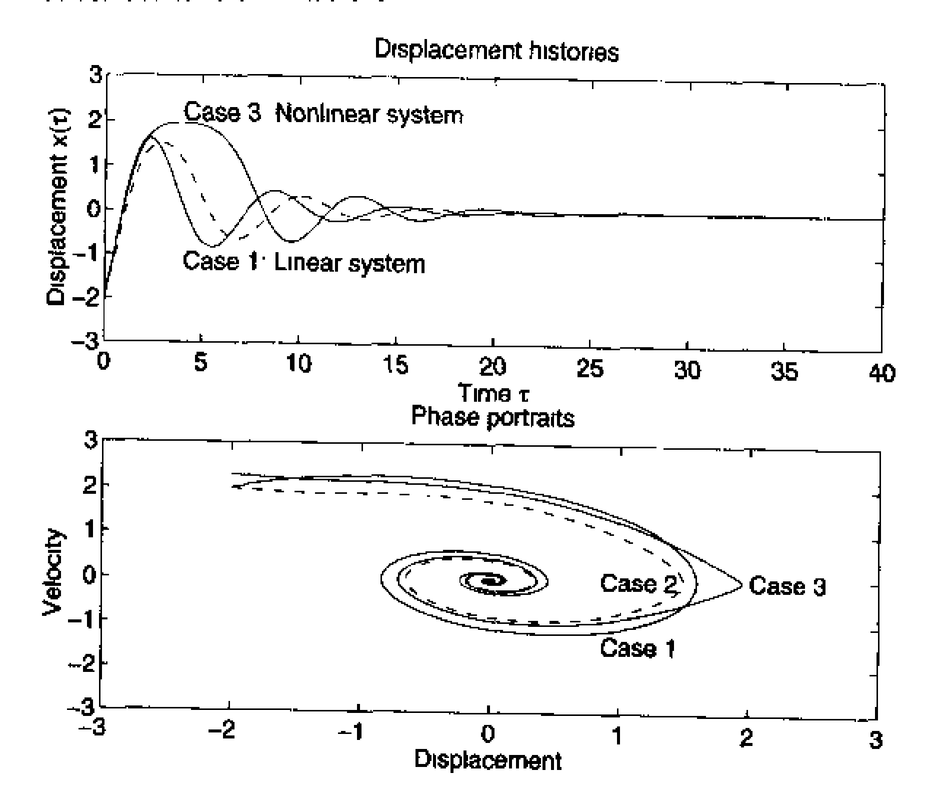


图 9.6 非线性阻尼振动固有响应的位移图和相平面图

第一种情况为线性系统,其阻尼振动周期在每次循环时实质上保持不变。第二和第三种情况均为非线性系统,其阻尼振动周期和阻尼因子在第 1 个循环与后继循环内的估算值有显著差异。正如图中第三种情况第 1 个周期所示,当振幅"大"时,非线性效果非常明显。

在第二种和第三种情况中,随着振幅变"小"而逐渐接近于线性系统(第一种情况)。

在相平面图中,与线性情况一样,非线性系统从初始条件开始的运动轨迹趋近于稳定的平衡位置(0.0,0.0)处。相平面图中振动曲线为螺旋形表明对应的非线性系统的振动为欠阻尼振动。在稳定的平衡位置附近的"小"振幅处,非线性系统应表现为线性系统。对初始条件不同的两种响应进行分析,可以看出在位移图上,第三种情况的第 1 个极值附近的曲线并不是典型的线性系统响应曲线。此时的曲线轨迹接近于系统的不稳定平衡位置(2.00,0.00)处,系统的振动受到影响。与线性系统不同,非线性系统可以有多个平衡位置,但并不是所有的平衡位置都稳定①。

① 参见 A H Nayfeh 和 B Balachandran,出处同上。

非线性阻尼系统　如图 9.7 所示的弹簧—质量系统，考虑系统有干摩擦，其方程为：

$$\frac{d^2x}{d\tau^2}+x+d\ \mathrm{signum}(dx/x\tau)=0 \tag{9.12}$$

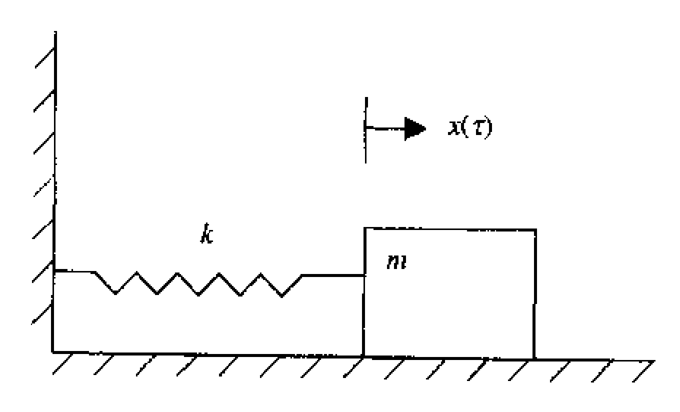


图 9.7　有干摩擦的弹簧—质量系统

式中常量 $d=\mu mg/k$，μ 为摩擦系数，mg 为物体的重量，k 为线性弹簧的弹性系数。干摩擦力是速度的分段常数函数，在式(9.12)中用 signum 表示。速度为正值时 signum 为 +1，速度为负值时 signum 为 −1。由于干摩擦力随速度的不同而呈现非线性变化，所以系统为一非线性系统。在系统的振动过程中，如果弹簧的弹性力不能克服干摩擦力时，系统将停止振动。即当：

$$\frac{dx}{d\tau}=0\qquad 且\ |x|\leqslant d \tag{9.13}$$

时，系统停止振动。式(9.12)表示的非线性系统具有多个平衡位置，这些平衡位置在相空间中形成连结点$(-d,0)$和点$(d,0)$的一条直线。

根据系统在 $dx/d\tau>0$ 范围内及 $dx/d\tau<0$ 范围内均为线性这一点，可以求出式(9.12)的封闭解[①]。但这里将用 ode45 求其数值解。首先引入变量：

$$x_1=x$$

$$x_2=\frac{dx}{d\tau}$$

式(9.12)变为一阶方程形式：

$$\begin{aligned}\frac{dx_1}{d\tau}&=x_2\\ \frac{dx_2}{d\tau}&=-x_1-d\ \mathrm{signum}(x_2)\end{aligned} \tag{9.14}$$

下面创建函数 FreeOscillation2 以表示式(9.13)和式(9.14)：

```
function xdot = FreeOscillation2(t,x,dummy,d)
if abs (x(1)) < = d & x(2) = = 0.0
 xdot = [0 0]';
else
 xdot = [x(2) - d * sign(x(2)) - x(1)]';
end
```

在 $d=0.86$，初始条件为(a)(3.0,0.0)及(b)(5.0,0.0)时，可用下列程序求得式(9.12)的数值解：

① 参见 D J Inman，出处同上，第 2 7 节；S S. Rao，出处同上，第 2.7 节。

```
d=0.86;
xo=[3 0 5.0];
vo=[0 0 0 0];
tspan=linspace(0,12,120);
options=odeset('AbsTol',[1e-3 1e-3]);
lintyp=['--''-'];
for i=1:2
 [t,x]=ode45('FreeOscillation2',tspan,[xo(i) vo(i)]',options,d);
 subplot(2,1,1);
 plot(t,x(:,1),lintyp(2*(i-1)+1:2*i));
 hold on
 subplot(2,1,2);
 plot(x(:,1),x(:,2),lintyp(2*(i-1)+1:2*i));
 hold on
 end
subplot(2,1,1)
xlabel('Time(\tau)');
ylabel('Displacement x(\tau)');
title('Displacement histories');
axis([0.0 12.0 -4.0 6.0]);
subplot(2,1,2)
xlabel('Displacement');
ylabel('Velocity');
title('Phase portraits');
text(2.5,0.5,'(3.0,0.0)');
text(4.5,0.5,'(5.0,0.0)');
axis([-4.0 6.0 -6.0 4.0]);
```

上述程序定义每种状态的绝对误差大于默认值 1.0e-6,从而加快计算速度。执行结果如图 9.8 所示。

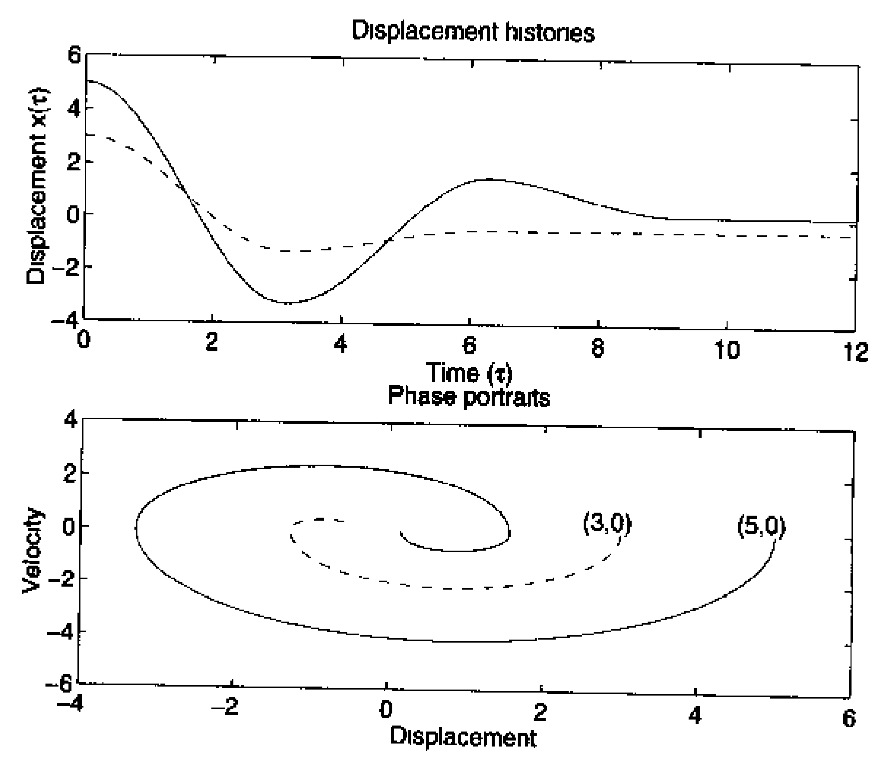


图 9.8 有干摩擦振动的固有响应位移曲线及相平面图

该系统在两个不同位置处停止振动,到达不同停止位置的时间也不同。这个例子说明非线性系统的长时间响应与初始条件有关,与之相反,一线性阻尼系统的渐近响应与初始条件无关。

9.2.4　线性系统与非线性系统的强迫振动

线性系统与非线性系统在谐波激励下的时域响应由 ode45 函数实现,对应的频域响应将由 fft 函数实现。参见第 5.6.4 节。

对于式(9.3)确定的系统,令所施力为简谐振动函数,即:

$$F(\tau)=X_o\cos(\Omega\tau) \tag{9.15}$$

其中 Ω 为无量纲激励频率,X_o 为所施力的幅值。选定激励频率和激励振幅,可以求出式(9.3)系统的稳态响应,即有:

$$\lim_{t\to\infty} x(t)$$

并检验在下列不同情况下响应的频谱:(1)$\hat{\alpha}=0$(线性系统),(2)$\hat{\alpha}\neq 0$(非线性系统)。至此得到下列函数:

```
xdot = [x(2); - 2 * zeta * x(2) - x(1) - AlphaHat * x(1)^3 + Xo * cos(Omega * t)];
```

在式(9.15)所表示的外力作用下,给定下列参数:(1)初始条件(0,0),(2)$\zeta=0.4$,(3)(非线性情况),(4)$\Omega=3.0$,(5)$X_o=5.0$。用 ode45 对式(9.3)进行数值积分。激励频率 Ω 选择为系统固有频率的三倍。保存上述数据以用于接下来的例子。程序为:

```
zeta = 0.4; AlphaHat = [0 1.5];
Omega = 3.0; Xo = 50 0;
tspan = linspace(0,30,6000);
sampint = tspan(2);
options = odesfet('RelTol',le - 8,'AbsTol',[le - 8 le - 8]);
for m = 1:2
 [t,x] = ode45('ForcedOscillatorl',tspan,[0 0]',options,zeta,AlphaHat(m),Omega,Xo);
 if m = = 1
  subplot(2,1,1);
  plot(t,x(:,1));axis([0 30 -8 8]);
  xlabel('\ tau');
  ylabel('x(\ tau)');
  title('Response of a linear system');
  yy = x(:,1);
  save 'c: \ path \ ForcedOscLin.txt'yy-ascii;
else
  subplot(2,1,2);
  plot(t,x(:,1));
  axis([0 30 -8 8])
  xlabel('\ tau');
  ylabel('x(\ tau)');
  title('Response of a nonlinear system');
  yy = x(:,1);
  save 'c: \ path \ ForcedOseNonLin.txt'yy-ascii;
  end
end
```

其中 *path* 由用户定义。

程序执行结果如图 9.9 所示。由图中看出，线性系统与非线性系统均到达稳态响应。该时间值与指标 $N_{start}=1\,600$ 有关。尽管两个稳态响应的周期都等于谐波强制函数的周期，但在频域图中可以清楚地区分两个响应的不同之处。

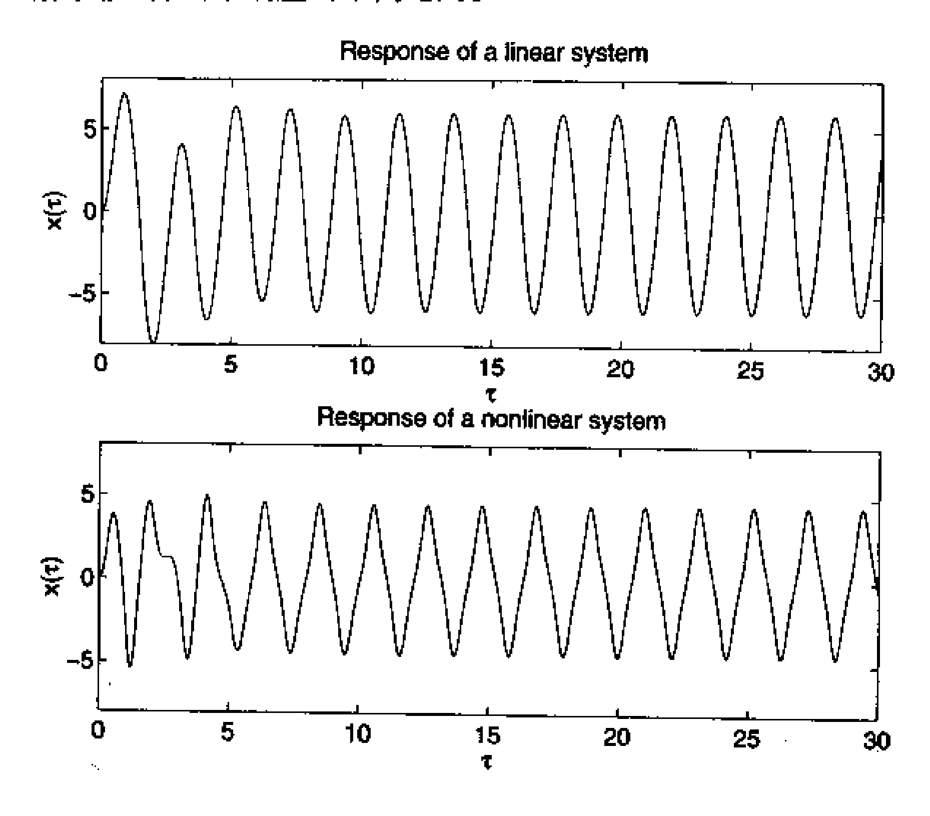


图 9.9 线性系统和非线性系统的响应图

为得到频域图中的数据，对 *ForcedOscLin.txt* 保存的线性系统时域图和 *ForcedOseNonlin.txt* 保存的非线性系统时域图的部分数据用 fft 函数求解。然后，利用第 5.6.4 节的结果可计算出 $\tau\geqslant 8$ 时每个信号的幅频谱，其过程如函数 AmplitudeSpectrum 所示：

```
function[f,amplitude] = AmplitudeSpectrum(AmplData,SamplingFreq,Nstart,N);
f = (SamplingFreq * (0:N - 1)/N) * 2.0 * pi;
amplitude = abs(fft(AmplData(Nstart:Nstart + N),N))/N;
```

对 *ForcedOscLin.txt* 与 *ForcedOscNonLin.txt* 中数据的采样速率为 $\tau_s=30/6\,000=0.005$，因此，采样频率(无量纲)为 $f_s=1/\tau_s=200$，对于基于激励频率的响应，这个值远远超出了必须达到的采样频率，结果不得不截短频谱图，这样仅能显示前 40 个值，而且需设置 $N_{start}=3\,200$，$N=2^{11}=2\,048$。程序清单为：

```
load'c:\path\ForcedOscLin.txt';
load'c:\path\ForcedOscNonLin.txt';
N = 2048;Nstart = 3200;Fs = 200;
[fLin,AmpLin] = AmplitudeSpectrum(ForcedOscLin,Fs,Nstart,N);
[fNonLin,AmpNonLin] = AmplitudeSpectrum(ForcedOscNonLin,Fs,Nstart,N);
subplot(2,1,1)
semilogy(fLin(1:40),2 * AmpLin(1:40));
xlabel('Frequency');
```

```
ylabel('Amplitude');
title('Response spectrum of a linear system');
text(3.1,10^4.5,'\Omega');
subplot(2,1,2);
semilogy(fNonLin(1:40),2*AmpNonLin(1:40));
v=axis;
xlabel('Frequency');
ylabel('Amplitude');
title('Response spectrum of a nonlinear system');
text(3.1,.5*v(4),'\Omega');
text(9.1,.2*v(4),'3\Omega');
```

其中 *path* 由用户定义。

该程序执行结果如图9.10所示。从图中可以看出，非线性系统的位移响应幅频谱在强制频率Ω及其整数倍处获得峰值，其余峰值由弹簧的三次非线性所致。对线性系统而言，只有一个频谱峰值，且与激励频率有关。上例说明非线性系统的响应可以有不同于激励频率的频谱成分。

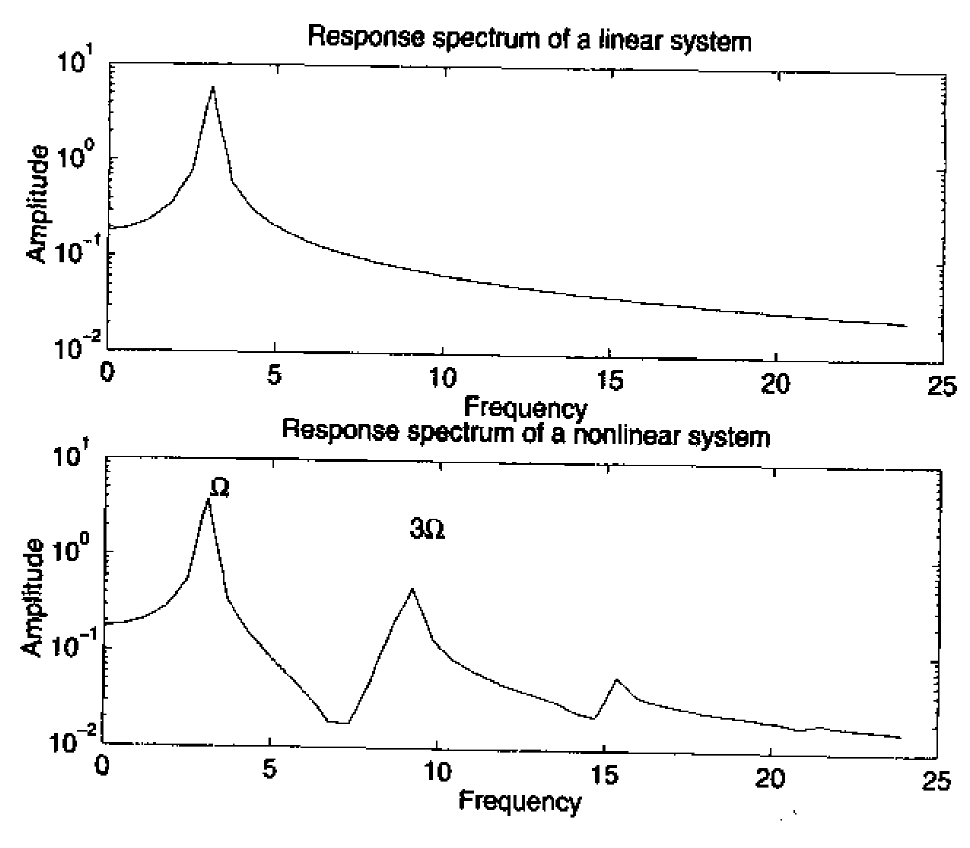


图9.10 线性系统和非线性系统响应的傅里叶分析

9.2.5 频率响应、阶跃响应及脉冲响应

下面说明怎样计算和显示线性系统的频率响应曲线。本节中介绍的原理与第9.2.4节介绍的原理基本相同，本节中给出了控制工具箱中的函数：

```
bode, tf, step, impulse, damp
```

的用法。一线性时不变系统(例如，常微分方程)的传递函数可表示为：

$$G(s)=\frac{N(s)}{D(s)}$$

其中 $N(s)$和 $D(s)$为复变量 s 的多项式。已知 $N(s)$和 $D(s)$ 系数数组时,函数:

```
sys = tf(N,D)
```

用于定义系统的传递系数。

频率响应函数:

$$G(s=i\omega)$$

可由

```
bode(tf(N,D),w)
```

计算并画出图形,图形包括 $G(i\omega)$的幅频特性及相频特性图,或者用:

```
[magnitude,phase] = bode(tf(N,D),w)
```

求出幅值及相位的数值解数组。数组 w 或为定义考虑范围内频率的最小值与最大值的两元素单元,或为一角频率数组。

函数 impuse 和 step 分别用于求线性时不变系统由静止到振动的脉冲响应与阶跃响应。

```
impulse(tf(N,D))
```

画出 tf 所描述系统的脉冲响应。

```
step(tf(N,D))
```

画出 $t=0$ 时系统的单位阶跃响应。关于这两个函数的其余用法可参考第 10 章。

函数:

```
[wn,zeta] = damp(tf(N,D))
```

用于求线性时不变系统传递函数的阻尼因子 ζ 和自然频率 ω_n。

为说明这些函数的用法,再考虑式(9.3)在 $\hat{\alpha}=0$ 时表示的系统。对该系统的传递函数做拉普拉斯变换①,则得到无量纲传递函数:

$$\frac{\bar{x}(s)}{X_o f(s)}=G(s)=\frac{1}{s^2+2\zeta s+1} \tag{9.16}$$

因此,有:

$$N(s)=1$$
$$D(s)=s^2+2\zeta s+1$$

定义 N 和 D 的数组分别为:

```
N = [0 0 1];
D = [1 2 * zeta 1];
```

频率响应——波特图 现用 tf 和 bode 函数将式(9.16)定义的传递函数用于构造频率响应曲线,程序为:

```
zeta = 0.2:0.2:1.0;
```

① 参见 L. Meirovitch,出处同上,附录 B。

```
omega = 0.0:0.01:3.0;
for i = 1:length(zeta)
 sys = tf([0 0 1],[1 2 * zeta(i) 1]);
 [mag,phas] = bode(sys,omega);
 subplot(2,1,1);
 plot(omega,mag(1,:));
 hold on;
 subplot(2,1,2)
 plot(omega,phas(1,:));
 hold on;
end
subplot(2,1,1)
xlabel('Frequency');
ylabel('Magnitude');
title('Amplitude response');
text(0.8,2.7,'\zeta = 0.2');
text(0.8,1.55,'\zeta = 0.4');
text(0.8,0.4,'\zeta = 1.0');
hold on,
plot([0.0 3 0],[1 0 1 0],'--');
subplot(2,1,2)
xlabel('Frequency');
ylabel('Phase(degrees)');
title('Phase response');
text(0.7, -15.0,'\zeta = 0.2');
text(0.5, -80.0,'\zeta = 1.0');
hold on;
plot([0.0 3.0]),[-90 -90,'--'];
plot([1.0 1.0]),[-200 0],'--'];
```

程序执行结果如图 9.11 所示。有些情况下无量纲传递函数的数值指放大倍数。

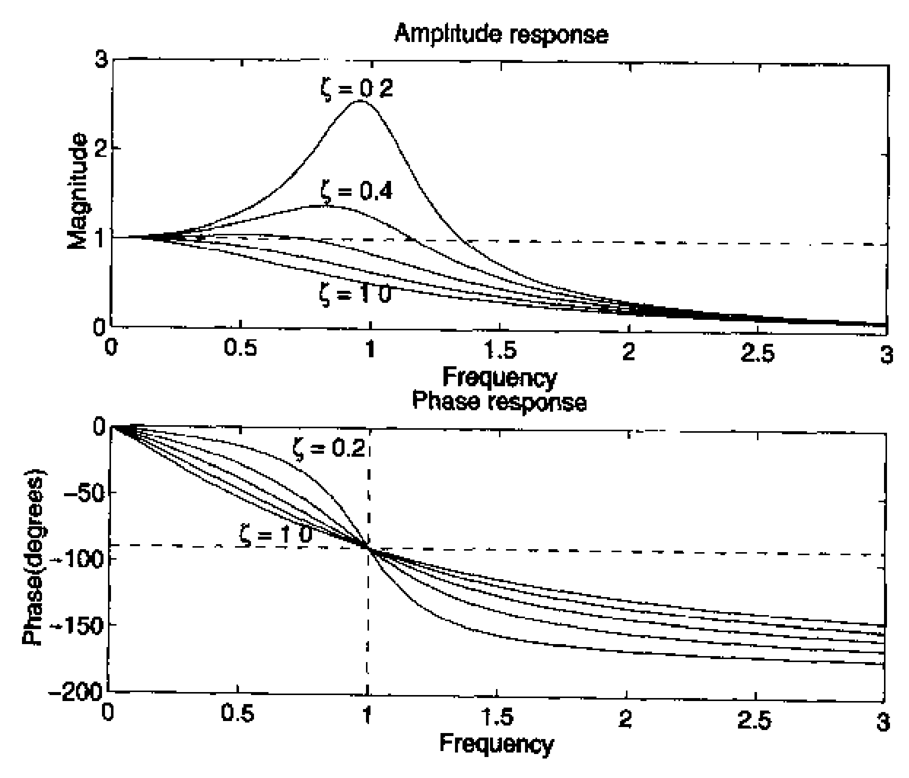


图 9.11　直接激励的弹簧—质量—阻尼器系统的幅频特性和相频特性

脉冲响应 下面求式(9.16)所描述系统的脉冲响应,分别对 $\zeta = 0.1$、1.0 和 3.0 所表示的欠阻尼系统、临界阻尼系统及过阻尼系统加以讨论,程序为:

```
eta = [0.2 1.0 3.0];tfinal = 30
tdata = linspace(0,tfinal,100);
for i = 1:length(zeta)
 xdata = impulse([0 0 1],[1 2 * zeta(i) 1],tdata);
 plot(tdata,xdata);
 hold on;
end
axis([0.0 30.0 -0 5 1.0];)
xlabel('Time \ tau');
ylabel('x( \ tau)');
title('Impulse response');
text(1 5,0 8,' \ zeta = 0 2');
text(10.5,0.08,' \ zeta = 3.0');
```

其执行结果如图 9.12 所示。通过对式(9.16)的时域模型在初始条件(0,1)下求积分,从而求得其自由振动响应,可获得图 9.12 的结果。

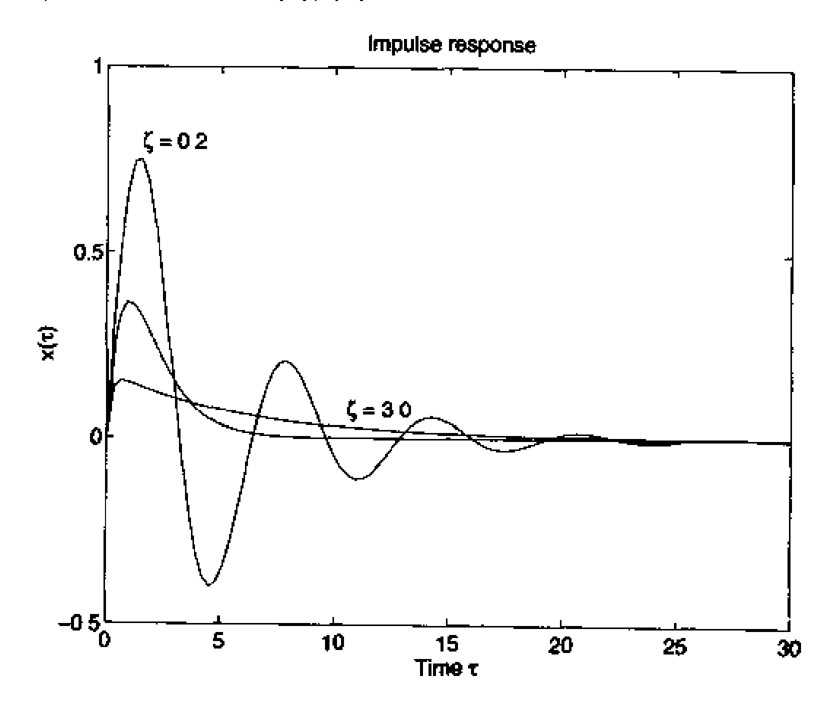


图 9.12 欠阻尼系统、临界阻尼系统、过阻尼系统的脉冲响应

阶跃响应 下面求式(9.16)系统的阶跃响应,分别对 $\zeta = 0.2$、1.0 及 3.0 所表示的欠阻尼系统、临界阻尼系统及过阻尼系统加以讨论。程序为:

```
zeta = [0.2 1.0 3.0];tfinal = 30;
tadat = linspace(0,tfinal,100);
for i = 1:length(zeta)
 xdata = step([0 0 1]),[1 2 * zeta(i) 1],tdata);
 plot(tdata,xdata);
 hold on;
end
xlabel('Time \ tau');
ylabel('x( \ tau)');
```

```
title('Step response');
text(5 0,1.10,'\zeta=0 2');
text(2.7,0.70,'\zeta=1.0');
text(5.0,0.55,'\zeta=3.0');
```

结果如图 9.13 所示。欠阻尼系统的响应在到达稳态之前，经过稳定位置上下摆动，而临界阻尼与过阻尼情况下却没有摆动。临界阻尼情况下，系统很快就到达稳定位置。

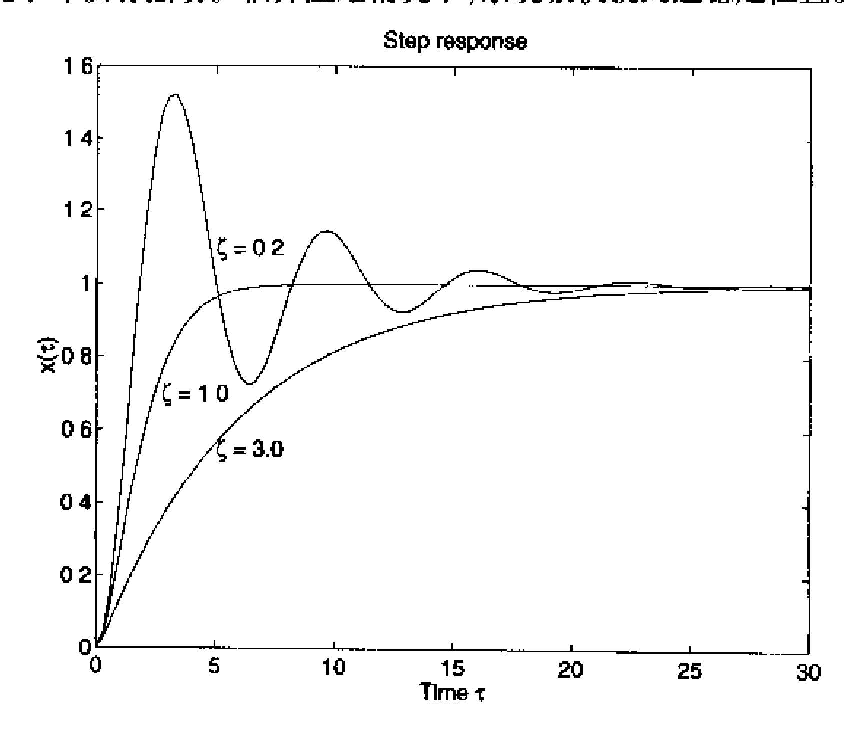


图 9.13 欠阻尼系统、临界阻尼系统及过阻尼系统的阶跃响应

ω_n 和 ζ 的估计 下面将求式(9.16)系统在 $\zeta=0.3$ 时的自然频率与衰减。首先令系统模型中的 $\zeta=0.3$，然后再用 damp 函数求其值。damp 函数由传递函数的极点求出阻尼因子以及与之相关的自然频率。实现语句为：

```
sys=tf([0 0 1],[1 2*0.3 1]);
[w,zeta]=damp(sys)
```

执行结果为 $\omega_n=1$ 和 $\zeta=0.3$。

迄今为止，只研究了这些函数用于单自由度系统的情况，它们还可用于研究多自由度系统，参考第 9.3 节。

9.3 多自由度系统

9.3.1 自由振动

下面分析三类不同问题，对每类问题都需要求出系统的特征值。

1. 主惯性矩问题

2. 转动刚体的稳定性问题

3. 多自由度系统的固有频率问题

主惯性矩问题 如图 9.14 所示一刚体有三个转动自由度。其相关的转动惯量矩阵有下述形式：

$$I_{rot}=\begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \tag{9.17}$$

其中各个惯性矩的定义参照图 9.14 的坐标系统。

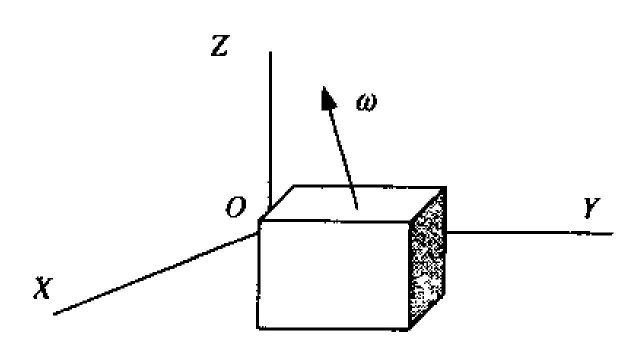


图 9.14 刚体在笛卡儿坐标轴中的转动

现定义一组新的正交坐标轴，使其转动惯量矩阵呈对角线分布，这组坐标轴被称为主轴，相应的惯性矩被称为主惯性矩。式(9.17)的特征值即为主惯性矩，其相应的特征向量则定义了主轴[①]。如下所述这两个量由 eig 函数求得。注意到矩阵的特征值之和等于矩阵的迹。矩阵的迹为矩阵对角线元素之和，由 trace 函数求得。而且令：

$$[I]=\begin{bmatrix} 150 & 0 & -100 \\ 0 & 250 & 0 \\ -100 & 0 & 500 \end{bmatrix}\text{kg m}^2 \tag{9.19}$$

程序如下：

```
Irot = [150 0 -100;0 250 0; -100 0 500];
[PrincipalDirections PrincipalMoments] = eig(Irot)
TraceIrot = trace(Irot)
TracePM = trace(PrincipalMoments)
```

其执行结果如下：

```
PrincipalDirections =
      0  -0.9665  -0.2567
  -1.0000        0        0
      0  -0.2567   0.9665
PrincipalMoments =
 250.0000       00        0
      0  123.4436        0
      0         0 526.5564
TraceIrot =
 900
TracePM =
 900
```

① 参见 D.T.Greenwood，出处同上。

尽管第一个特征向量只与第一个特征值有关，第二个特征向量只与第二个特征值有关，并可依此类推，但特征值(这里指主惯性矩)并没有一定的顺序，这点是 eig 函数所返回结果的特点。分析上述转动惯量矩阵发现，y 轴为一主轴，因此其中一个特征值等于 I_{yy}。主方向矩阵定义一个方向余弦矩阵将(x, y, z)坐标轴转换为主轴。

刚体的稳定性　参照图 9.14 所示的刚体，设 I_1、I_2 和 I_3 分别表示沿物体各个主轴方向定义的 x 轴、y 轴及 z 轴的第二质量惯性矩。即前例中的主惯性矩，ω_1、ω_2 及 ω_3 分别表示各个主轴的角速度，M_1、M_2 及 M_3 分别表示各轴的外惯性矩。其运动方程即著名的欧拉(Euler)方程，具有下述形式[①]：

$$\begin{aligned} I_1\dot{\omega}_1 + (I_3 - I_2)\omega_2\omega_3 &= M_1 \\ I_2\dot{\omega}_2 + (I_1 - I_3)\omega_3\omega_1 &= M_2 \\ I_3\dot{\omega}_3 + (I_2 - I_1)\omega_1\omega_2 &= M_3 \end{aligned} \tag{9.19}$$

其中：

$$\dot{\omega}_j = \frac{d\omega_j}{dt} \qquad j = 1,2,3$$

在自由振动情况下，即 $M_1 = M_2 = M_3 = 0$ 时，ω_i 不随时间而变化，其解有三种类型，这些解被称做常数解，如下所示：

1. $(\omega_{10} \neq 0, \omega_{20} = 0, \omega_{30} = 0)$

2. $(\omega_{10} = 0, \omega_{20} \neq 0, \omega_{30} = 0)$

3. $(\omega_{10} = 0, \omega_{20} = 0, \omega_{30} \neq 0)$

每种解对应一个主轴方向的纯旋转动运动。这里只关心怎样求出三种振动的稳定位置。为此令 $\xi_j(j = 1,2,3)$表示系统各轴上所加干扰，则有：

$$\begin{aligned} \omega_1(t) &= \omega_{10} + \xi_1(t) \\ \omega_2(t) &= \omega_{20} + \xi_2(t) \\ \omega_3(t) &= \omega_{30} + \xi_3(t) \end{aligned} \tag{9.20}$$

将式(9.20)代入式(9.19)，并假定干扰量数值很“小”，则可对式(9.19)线性化[②]，从而分析相关的特征值问题。其结果如下：

$$\begin{bmatrix} 0 & (I_3 - I_2)\omega_{30}/I_1 & (I_3 - I_2)\omega_{20}/I_1 \\ (I_1 - I_3)\omega_{30}/I_2 & 0 & (I_1 - I_3)\omega_{10}/I_2 \\ (I_2 - I_1)\omega_{20}/I_3 & (I_2 - I_1)\omega_{10}/I_3 & 0 \end{bmatrix} \begin{Bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{Bmatrix} = \lambda \begin{Bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{Bmatrix} \tag{9.21}$$

在式(9.21)的三个特征值中，如果有一个或更多个具有正实部，则干扰值将增加，对应的振动将不稳定。因式(9.21)的迹为 0，所以特征值之和也将为 0。令刚体的 $I_1 = 150\ \text{kg m}^2$，$I_2 = 50\ \text{kg m}^2$，$I_3 = 300\ \text{kg m}^2$，分析三种常数解情况下每种解的稳定性。下面的程序用于求沿最大惯性轴、最小惯性轴及另一惯性轴加于旋转振动上的干扰量的特征值 λ：

```
I = [150 50 300];
omega10 = [1 0 0];
```

① 参见 D.T Greenwood，出处同上，第 392 页；F C Moon，出处同上，第 192 页。

② 参见 A.H Nayfeh 和 B.Balachandran，出处同上。

```
omega20 = [0 1 0];
omega30 = [0 0 1];
for i = 1:length(omega10)
  A = [0(I(3) - I(2)) * omega30(i)/I(1)(I(3) - I(2)) * omega20(i)/I(1);...
      (I(1) - I(3)) * omega30(i)/I(2)0(I(1) - I(3)) * omega10(i)/I(2);...
      (I(2) - I(1)) * omega20(i)/I(3)(I(2) - I(1)) * omega10(i)/I(3) 0];
  fprintf(1,'\nCase %3d:Eigenvalues\n',i);
  lambda = eig(A)
  SumLambda = sum(lambda)
end
```

运行结果在 MATLAB 命令窗口中显示下列数据：

```
Case  1:Eigenvalues
lambda =
   1
  -1
   0
sum_lambda =
  0
Case  2:Eigenvalues
lambda =
     0 + 0.7454i
     0 - 0.7454i
     0
sum_lambda =
0
Case  3:Eigenvalues
lambda =
     0 + 2.2361i
     0 - 2.2361i
     0
sum_lambda =
  0
```

对上述三种情况，每一种的特征值总有一个为 0，而且特征值之和总为 0。对第一种情况，其初始转动方向为沿惯性矩中间值的坐标轴方向，有一个特征值具有正实部，说明振动是不稳定的，对应的物理振动则是摇摆的振动。对第二种情况，其初始转动振动方向为沿惯性矩最小的坐标轴方向，有两个特征值组成一对纯虚数。对第三种情况，其初始旋转振动方向为沿惯性矩最大的坐标轴方向，有两个特征值组成一对纯虚数。在第二种和第三种情况下，因为没有一个特征值有正实部，所以系统所受的干扰不会增加，因此这两种情况下系统的振动是稳定的。如进一步用数值积分方法分析式(9.19)的后两种情况，可以发现振动轨迹为三维空间中的环形轨迹，用于定义三维空间的三个状态为 ω_1、ω_2 及 ω_3。这些轨迹位于 Poinsot 椭球面上①。

三自由度系统的振动模态及其固有频率　一系统如图 9.15 所示，x_1, x_2, x_3 为距离系统静止平衡位置的位移，系统的控制方程为：

① 参见 D T. Greenwood，出处同上，第 8 4 节。

$$\begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \begin{Bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \\ \ddot{x}_3 \end{Bmatrix} + \begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & (k_1+k_2)-k_2 & \\ 0 & -k_2 & k_2 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} \tag{9.22}$$

其中，

$$x_j = \frac{d^2 x_j}{dt^2}$$

相应的特征值问题描述如下：

$$\begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & (k_1+k_2)-k_2 & \\ 0 & -k_2 & k_2 \end{bmatrix} \begin{Bmatrix} v_1 \\ v_2 \\ v_3 \end{Bmatrix} = \lambda \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \begin{Bmatrix} v_1 \\ v_2 \\ v_3 \end{Bmatrix} \tag{9.23}$$

其中 $\lambda = \omega^2$。

假定求特征值及其相关特征向量的程序如下：

```
k = [100  - 100 0; - 100 150  - 50;0  - 50 50];
m = [100 0 0;0 100 0;0 0 100];
[VibrationModes, Eigenvalues] = eig(k,m)
```

执行结果为：

```
VibrationModes =
  0.5774  0.5774   - 0.5774
 - 0.7887  0.5774   - 0.2113
  0.2113  0.5774    0.7887
Eigenvalues =
 2.3660     0      0
   0  0.0000      0
   0      0  0.6340
```

观察图(9.15)所示系统，发现质量为 m_1, m_2, m_3 的小车两端并未受到弹簧的弹性恢复力，因此，该系统具有刚体模式。该模式中所有物体以相同方向移动相同的值。这一点在对应的振动模式中已体现出来，如振动模式矩阵中第 2 列元素所示。这种情况下弹簧既没有伸长也没有缩短，振动的特征值为 0。

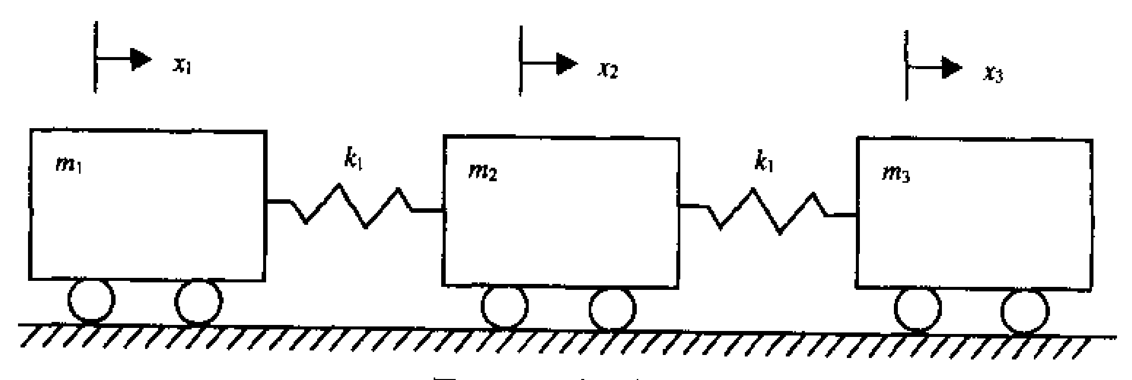


图 9.15　三自由度系统

如果一方阵有一零特征值，则矩阵的行列式为 0。为证实矩阵是否有零特征值，可求出矩阵的秩。矩阵的秩是矩阵中行列式不为 0 的最大子方阵的阶，可由函数 rank 求得。

下列程序求式(9.22)所示的刚性矩阵是否有一零特征值：

```
k = [100 - 100 0; - 100 150  - 50;0  - 50 50];
```

```
mk = rank(k);
[m n] = size(k);
disp(['Number of zero eigenvaules is'num2str(m - mk,2)])
```

执行结果产生下列输出：

```
Number of zero eigenvalues is 1
```

如果刚性矩阵的秩为2,说明在这个(3×3)阶刚性矩阵中有一个(2×2)阶矩阵的行列式不为0。

9.3.2 强迫振动及减振器

如图9.16所示,一受外力 $F(t)$作用的两自由度系统,其中质量 m_2、弹簧 k_2 及阻尼器 c_2 组成第二级系统,质量 m_1、弹簧 k_1 及阻尼器 c_1 组成主系统。第二级系统称做减振器,它加在受外力作用的主系统之上以减弱主系统的振动。

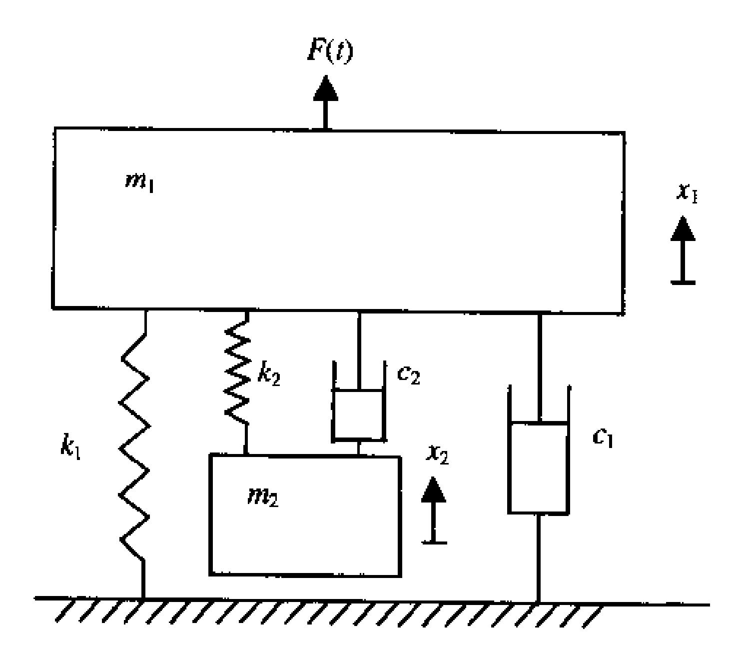


图9.16 受外力 $F(t)$作用的两自由度系统

该系统的传递函数如下所示：

$$\begin{aligned}\frac{\bar{x}_1(s)}{F(s)} &= \frac{m_2 s^2 + c_2 s + k_2}{D(s)} \\ \frac{\bar{x}_2(s)}{F(s)} &= \frac{c_2 s + k_2}{D(s)}\end{aligned} \tag{9.24}$$

其中，

$$\begin{aligned}D(s) = m_1 m_2 s^4 &+ [(c_1 + c_2)m_2 + c_2 m_1]s^3 + [(k_1 + k_2)m_2 + k_2 m_1 + c_1 c_2]s_2 \\ &+ (k_1 c_2 + k_2 c_1)s + k_1 k_2\end{aligned} \tag{9.25}$$

由于这两个传递函数将被多次应用,所以下面创建函数 Transferab：

```
function sys = Transferab(m,k,c)
N = {[m(2) c(2) k(2)];[c(2) k(2)]};
D = [m(1) * m(2)((c(1) + c(2)) * m(2) + c(2) * m(1))..
    ((k(1) + k(2)) * m(2) + k(2) * m(1) + c(1) * c(2))  .
    (k(1) * c(2) + c(1) * k(2))k(1) * k(2)];
```

```
sy = tf(N,D);
```

脉冲响应 假定：

$m_1 = 50\ \text{kg}$ $k_1 = 200\ \text{N/m}$ $c_1 = 10\ \text{Ns/m}$

$m_2 = 10\ \text{kg}$ $k_2 = 40\ \text{N/m}$ $c_2 = 6\ \text{Ns/m}$

在 $t=0$ 时，给质量为 m_1 的物体施加一单位脉冲，现在求质量为 m_1 和 m_2 的两物体的响应。程序如下：

```
m = [50 10];
k = [200 40];
c = [10 6];
[y,t] = impulse(Transferab(m,k,c),20)
subplot(2,1,1);
plot(t,y(:,1,1));
ylabel('x _ 1(t)');
title('Impulse response of m _ 1');
subplot(2,1,2);
plot(t,y(:,2,1));
xlabel('Time t');
ylabel('x _ 2(t)');
title('Impulse response of m _ 2');
```

程序执行结果如图9.17所示。从图中看出，两个物体在初始时瞬态运动是不同的。尽管脉冲加于物体 m_1 上，而物体 m_2 的响应幅值在初始时却大于 m_1 的响应幅值。经过大约7秒钟左右，两个物体的振动周期看起来相等。

通过将 impulse 函数换为 step 函数可以类似方法得到该系统的阶跃响应。

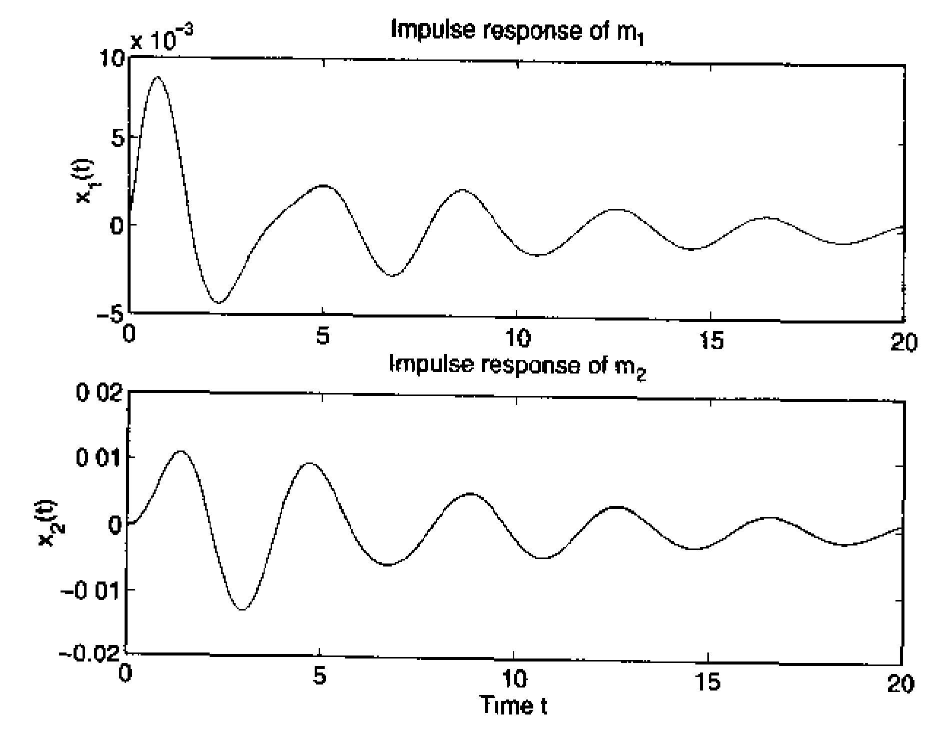


图9.17 在 m_1 上加一脉冲时两自由度系统的脉冲响应

减振器 现在求 m_1 上施加一外力时，物体 m_1 和 m_2 位移响应的频率响应函数。这类分析方

法用于减振器的设计。设计减振器的目的是减弱主要物体在干扰频率下的位移响应[①]。利用前例中相同的系统参数,程序代码为:

```
m=[50 10];
k=[200 40];
c=[10 6];
omega=0 0:0.005:4
for i=1:2
 if i==1
  sys=tf([1],[m(1)c(1)k(1)]);
  [mag phas]=bode(sys,omega);
plot(omega,mag(1,:),'--');
  hold on;
 else
  sys=Transferab(m,k,c);
  [mag,phas]=bode(sys,omega);
  plot(omega,mag(1,:));
 end
end
xlabel('Excitation frequency(rad/s)')
ylabel('|x_1|')
text(2,1,0.045,'No vibration absorber')
text(0.3,0.02,'With vibration absorber')
```

程序执行结果如图 9.18 所示。由图上看出 m_1 的响应曲线在频率为 2 rad/s 附近衰减,这个值为主系统的无阻尼自然频率,即 m_2 的自然频率。有减振器的系统有两个自由度,相应的系统表现为有两个谐振频率。这里没有进一步采取得到优化参数的措施。

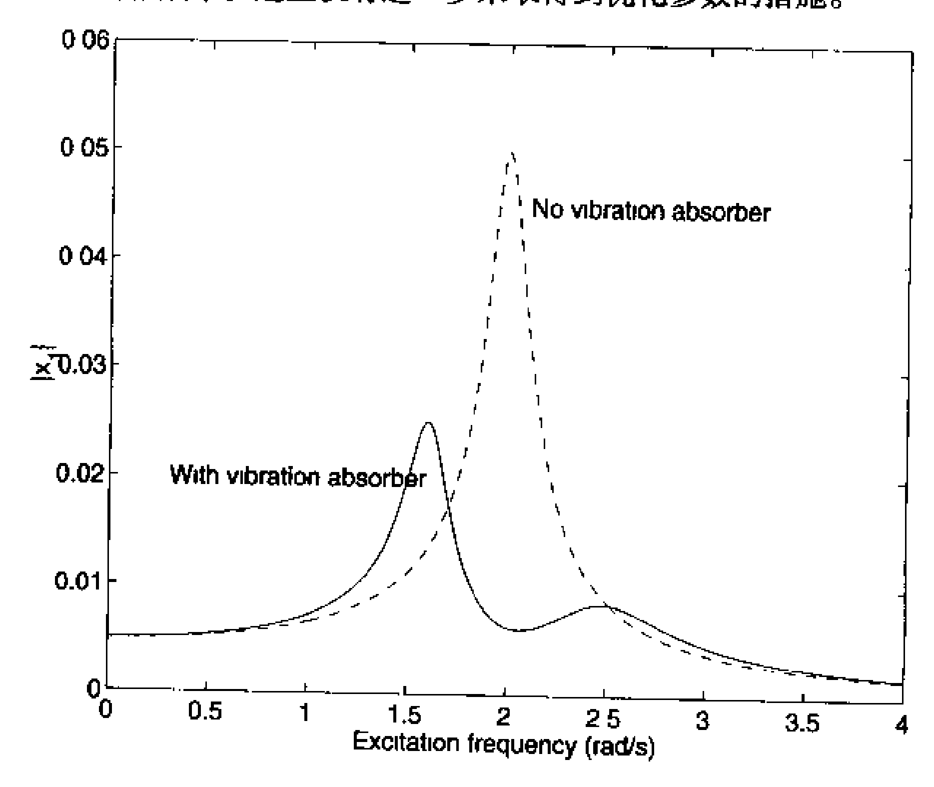


图 9.18 有减振器和没有减振器时第一级质量系统的响应幅度

① 参见 D J Inman,出处同上,第 5.3 节;S S Rao,出处同上,第 9 10 节;B H Tongue,出处同上,第 4 4 节。

9.4 细梁的振动[①]

9.4.1 等截面梁

一梁长度为 L,截面积为 A,惯性矩为 I,密度为 ρ,弹性模量为 E,受一动力 $F(x,t)$作用,其横向位移为 $w'(x,t)$时的无量纲振动控制方程为:

$$\frac{\partial^4 w}{\partial u^4}+\frac{\partial^2 w}{\partial \tau^2}=F_o(u,\tau) \tag{9.26}$$

其中 $u=x/L,\tau=c_0t,\mathrm{w}(\mathrm{u},\tau)=w'(x/L,c_0t)/L,F_o(u,\tau)=F(x/L,c_0t)L^3/EI,c_0=rc_b/L^2,r^2=I/A$ 和 $c_b^2=E/\rho$。

假设梁具有下述一般边界条件:

u=0:

$$\frac{\partial^3 w}{\partial u^3}=-\alpha_1 w$$

$$\frac{\partial^2 w}{\partial u^2}=\gamma_1\frac{\partial w}{\partial u} \tag{9.27a}$$

u=1:

$$\frac{\partial^3 w}{\partial u^3}=\alpha_2 w+\frac{M_o}{m_o}\frac{\partial^2 w}{\partial \tau^2}$$

$$\frac{\partial^2 w}{\partial u^2}=-\gamma_2\frac{\partial w}{\partial u} \tag{9.27 b}$$

其中 M_0 为附加质量值。$M_0=\rho AL$ 为梁的质量,

$$\alpha_j=\frac{k_jL^3}{EI}\qquad \gamma_j=\frac{\beta_jL}{EI}\qquad j=1,2$$

分别为弹簧的无量纲线性常数与扭转常数,k_j 为弹簧的线性常数,β_j 为扭转弹簧常数。

假定初始条件为 0,式(9.26)系统在式(9.27a,b)的一般边界条件下的解为:

$$w(u,\tau)=\sum_{n=1}^{\infty}\frac{W_n(u)}{\Omega_n^2N_n}\int_0^1\int_0^{\tau}F(u,\tau')W_n(u)\sin[\Omega_n(\tau-\tau')]\mathrm{d}\tau'\mathrm{d}u \tag{9.28}$$

其中,

$$W_n(u)=A'_nU(\Omega_nu)+B'_nV(\Omega_nu)+C'_nS(\Omega_nu)+D'_nT(\Omega_nu) \tag{9.29}$$

为 Ω_n 的模型函数,

$$U(u)=[\cos(u)+\cosh(u)]/2$$

$$V(u)=[\sin(u)+\sinh(u)]/2$$

$$S(u)=[\cosh(u)+\cos(u)]/2$$

$$T(u)=[\sinh(u)+\sin(u)]/2$$

---

① 本节由 Edward B. Magrab 所写,部分内容摘自 E B. Magrad, *Vibrations of Elastic Structural Members*, Sijthoff & Noordhoff, The Netherlands, 1979 年,第 4 章。

和

$$N_n = \frac{1}{4\Omega_n}\{G_1 + \cosh 2\Omega_n(G_2\tanh 2\Omega_n + G_3) + \cosh\Omega_n[(G_4 + G_6)\tanh\Omega_n + G_5 + G_7]\}\frac{M_0}{m_0} + W_n^2(1) \tag{9.30}$$

在式(9.30)中，

$$G_1 = \Omega_n(A_n'^2 + C_n'^2 - 2B'_nD'_n) + B'_nC'_n - 3A'_nD'_n + 0.25[(A'_n - C'_n)^2 - (B'_n - D'_n)^2]\sin 2\Omega_n - 0.5(B'_n - D'_n)(A'_n - C'_n)\cos 2\Omega_n$$

$$G_2 = 0.25[(A'_n + C'_n)^2 + (B'_n + D'_n)^2]$$

$$G_3 = 0.5(B'_n + D'_n)(A'_n + C'_n)$$

$$G_4 = (A_n'^2 - C_n'^2 + D_n'^2 - B_n'^2)\cos\Omega_n$$

$$G_5 = (A_n'^2 - C_n'^2 - D_n'^2 + B_n'^2)\sin\Omega_n$$

$$G_6 = 2(A'_nB'_n - C'_nD'_n)\sin\Omega_n$$

$$G_7 = 2(A'_nD'_n - C'_nB'_n)\cos\Omega_n$$

另外，频率系数 $\Omega_n = \sqrt{\omega_n/c_0}$，其中 $\omega_n = 2\pi f_n$ 为自然角频率，Ω_n 为下式的一个根。

$$\Lambda(\Omega_n) = z_1[\cos\Omega_n\tanh\Omega_n + \sin\Omega_n] + z_2[\cos\Omega_n\tanh\Omega_n - \sin\Omega_n] - 2z_3\sin\Omega_n\tanh\Omega_n + z_4(\cos\Omega_n - 1/\cosh\Omega_n) + z_5(\cos\Omega_n + 1/\cosh\Omega_n) + 2z_6\cos\Omega_n = 0 \tag{9.31}$$

式中，

$$z_1 = [b_{1n}b_{2n}(a_{1n} + a_{2n}) + (b_{1n} - b_{2n})]$$

$$z_2 = [a_{1n}a_{2n}(b_{1n} - b_{2n}) - (a_{1n} + a_{2n})]$$

$$z_3 = (a_{1n}a_{2n} + b_{1n}b_{2n})$$

$$z_4 = (1 - a_{1n}a_{2n}b_{1n}b_{2n})$$

$$z_5 = (a_{2n}b_{2n} - a_{1n}b_{1n})$$

$$z_6 = (a_{1n}b_{2n} - a_{2n}b_{1n})$$

和

$$a_{1n} = \frac{\alpha_1}{\Omega_n^3} \qquad b_{1n} = \frac{\gamma_1}{\Omega_n}$$

$$a_{2n} = \frac{\alpha^2}{\Omega_n^3} - \frac{M_0}{m_0}\Omega_n \qquad b_{2n} = \frac{\gamma_2}{\Omega_n}$$

表 9.3 列出了 A'_n, B'_n, C'_n, D'_n 的表达式，这些表达式为边界条件的函数。选择边界条件可涉及大量特殊情况。获得这些特殊情况的方法是通过将参数 $a_{1n}, a_{2n}, b_{1n}, b_{2n}$ 设为 0 或 ∞。表 9.4 中总结了几种较一般的边界条件，还有 A'_n, B'_n, C'_n, D'_n 与 $\Lambda(\Omega_n)$ 及对应的关系。

表 9.3　系数 A'_n, B'_n, C'_n, D'_n 的定义

情况	a_n'	b_n'	A_n'	B_n'	C_n'	D_n'
1	$0 \leqslant a_{1n} < \infty$	$0 \leqslant b_{1n} < \infty$	E_n	1	b_{1n}	$-a_{1n}E_n$
2	$0 < a_{1n} \leqslant \infty$	$0 \leqslant b_{1n} < \infty$	$-En/a_{1n}$	1	b_{1n}	E_n
3	$0 \leqslant a_{1n} < \infty$	$0 < b_{1n} \leqslant \infty$	E_n	$1/b_{1n}$	1	$-a_{1n}E_n$
4	$0 < a_{1n} \leqslant \infty$	$0 < b_{1n} \leqslant \infty$	$-En/a_{1n}$	$1/b_{1n}$	1	E_n

情况	E_n
1	$\dfrac{a_{2n}V(\Omega_n)+(a_{2n}b_{1n}-1)S(\Omega_n)-b_{1n}T(\Omega_n)}{V(\Omega_n)-(a_{1n}+a_{2n})U(\Omega_n)+a_{1n}a_{2n}T(\Omega_n)}$
2	$\dfrac{a_{2n}V(\Omega_n)+(a_{2n}b_{1n}-1)S(\Omega_n)-b_{1n}T(\Omega_n)}{-(1/a_{1n})V(\Omega_n)+(a_{2n}/a_{1n}+1)U(\Omega_n)-a_{2n}T(\Omega_n)}$
3	$\dfrac{a_{2n}/b_{1n}V(\Omega_n)+(a_{2n}-1/b_{1n})S(\Omega_n)-T(\Omega_n)}{V(\Omega_n)-(a_{2n}+a_{1n})U(\Omega_n)a_{1n}a_{2n}T(\Omega_n)}$
4	$\dfrac{a_{2n}/b_{1n}V(\Omega_n)+(a_{2n}-1/b_{1n})S(\Omega_n)-T(\Omega_n)}{-(1/a_{1n})V(\Omega_n)+(a_{2n}/a_{1n}+1)U(\Omega_n)-a_{2n}T(\Omega_n)}$

表 9.4　表 9.3 给出的一般解的几种特殊情况

边界条件										
$u=0$	$u=1$	M_0	a_{1n}	b_{1n}	a_{2n}	b_{2n}	A_n'	B_n'	C_n'	D_n'
ss	ss	0	∞	0	∞	0	0	1	0	-1
c	c	0	∞	∞	∞	∞	0	0	1	$-S(\Omega_n)/T(\Omega_n)$
c	f	M_0	∞	∞	$-(M_0/m_0)\Omega_n$	0	0	0	1	$-\dfrac{T(\Omega_n)+(\Omega_nM_0/m_0)S(\Omega_n)}{U(\Omega_n)-(\Omega_nM_0/m_0)T(\Omega_n)}$
c	f	0	∞	∞	0	0	0	0	1	$-T(\Omega_n)/U(\Omega_n)$
c	ss	0	∞	∞	∞	0	0	0	1	$-S(\Omega_n)/T(\Omega_n)$

边界条件[①]

$u=0$	$u=1$	$\Lambda(\Omega_n)$	表 9.3 的情况
ss	ss	$\sin(\Omega_n)$	2
c	c	$\cos(\Omega_n)\cosh(\Omega_n)-1$	4
c	f	$(M_0/m_0)\Omega_n[\cos(\Omega_n)\sinh(\Omega_n)]$ $-\sin(\Omega_n)\cosh(\Omega_n)]+\cos(\Omega_n)\cosh(\Omega_n)+1$	4
c	f	$\cos(\Omega_n)\cosh(\Omega_n)+1$	
c	ss	$\tanh(\Omega_n)-\tan(\Omega_n)$	4

① ss = 简单支撑状态，c = 夹紧状态，f = 自由状态。

如梁受一脉冲作用，且 $u=\xi(0<\xi 1<)$时，有：

$$F_v(u,\tau)=\delta(u-\xi)\delta(\tau)$$

则式(9.28)变为：

$$w(u,\tau)=\sum_{n=1}^{\infty}\frac{W_n(u)W_n(\xi)}{\Omega_n^2N_n}\sin(\Omega_n\tau) \tag{9.32}$$

因为参数 a_{1n}、a_{2n}、b_{1n}及 b_{2n}的取值范围从 0 到 ∞，所以必须考虑这四个参数取得极值的 16 种不同组合情况。由于 MATLAB 不能表达极限值，因此对每种情况需对式(9.31)进行巧妙的代数处理以得到每种表达式的正确表示。这里采用表 9.5 中的方法识别 16 种情况的每一种。同时还能正确识别包含某表达式的函数名，以 b2、a2、b1 及 a1 标识的四位数值前加上字符串'beam'，文件名即为['beam' 'b2' 'a2' 'b1' 'a1']，取值为 0 或 1。取值为 1 时表示该参数可由一大于 0 的值增加到无穷，取值为 0 时表示该参数可增加到小于 0 的一个极大值。本章末尾附录 B 列出了函数 *beamxxxx*，包含 $\Lambda(\Omega_n)=0$ 时的 16 种表达式，用于求自然频率系数 Ω_n。同样必须求出正确的 $W_n(u)$及包含它的文件名。但由于表达式中没有 b_{2n}，所以只有 8 个函数，用于计算模态的函数名为['mode''0''a2''b1''a1']。这样总共创建 24 个函数。而且，由于已求出每种特殊情况的 A_n'，B_n'，C_n'，D_n'，所以可用函数 mode0xxx 计算 Nn 的值。函数beamxxxx和mode0xxx在本章末尾附录 B 中列出。

表 9.5 式(9.35)的 16 种特殊情况的函数命名方法

b2	0	0	0	0	0	0	0	0
a2	0	0	0	0	∞	∞	∞	∞
b1	0	0	∞	∞	0	0	∞	∞
a1	0	∞	0	∞	0	∞	0	∞
b2 a2 b1 a1	0000	0001	0010	0011	0100	0101	0110	0111
b2	∞	∞	∞	∞	∞	∞	∞	∞
a2	0	0	0	0	∞	∞	∞	∞
b1	0	0	∞	∞	0	0	∞	∞
a1	0	∞	0	∞	0	∞	0	∞
b2 a2 b1 a1	1000	1001	1010	1011	1100	1101	1110	1111

例 9.1 正常模态输出与脉冲响应输出

下面编写程序并创建用到的函数，该程序具有下述功能：

1. 对梁两端的边界条件给用户提供多种组合选择。
2. 计算出前 20 个自然频率系数并输出前 4 个正常化模态及其对应的 Ω_n/π。
3. 用求得的前 20 个自然频率系数计算并输出梁的位移，条件为对梁施加 τ 脉冲力，取0.1 到 0.8 的 8 个值。

为实现上述功能，创建三个函数，其代码在本章末尾的附录 B 中第 24 个函数后列出。第一个为 GetBC，用于接受用户输入的边界条件；第二个函数为 BeamEqn，用于将所选边界条件转换为文件名以访问其正确的 $\Lambda(\Omega_n)$和 $W_n(u)$；第三个函数为 N_n，用于计算 N_n。为方便起见，仍用第 4.3.2 节及图 5.1 介绍的 Manyzeros 函数，此函数用于求多次方根，其实现代码如下：

```
function nRoots = ManyZeros(FileName,n,xstart,toler,increment,g)
x = xstart;
dx = increment;
for m = 1:n
```

```
  sl = sign(feval(FileName,x,g));
  while dx/x > toler
   if sl ~ = sign(feval(FileName,x + dx,g))
    dx = dx/2
   eles
    x = x + dx;
   end
  end
  nRoots(m) = x;
  dx = increment;
  x = 1.05 * x;
end
```

程序为：

```
nfnum = 20; u = 0:0.02:1; xi = 0.4;
tau = 0.1:.1:.8;
gg = GetBC
[nffile,modefile] = BeamEqn(gg(7:10));
nfcoeff = ManyZeros(nffile,nfnum,0.4,1e - 6,.4,gg(1:6));
lit = ['First','Second','Third','Fourth'];
for j = 1:nfnum
 wnx(j) = feval(modefile,nfcoeff(j),gg(1:6),xi);
 [mode,c] = feval(modefile,nfcoeff(j),gg(1:6),u);
 wnu(j,:) = mode;
 nn(j) = c;
 if j< =4
  subplot(2,2,j)
  plot(u,mode/max(abs(mode)),'k',[0,1],[0,0],'k-')
  axis([0,1, - 1,1])
  xlabel('Beam position')
  ylabel('Relative amplitude')
  title([lit(j * 6 - 5:j * 6),'mode: \ Omega ∧ pi = 'num2str(nfcoeff(j)/pi,6)])
 end
end
wtau = (wnu'. * repmat(wnx./(nn. * nfcoeff.^2),length(u),1)) * sin(nfcoeff' * tau);
figure(2)
for k = 1:8
 if k<5
  subplot(4,2,2 * k - 1)
 else
  subplot(4,2,2 * k - 8)
 end
 plot(u', - wtau(:,k))
 if k= =4|k= =8
  xlabel('Beam position')
 end
 text(.1,.15,[' \ tau = 'num2str(tau(k),3)])
 axis([0 1 - .2 .2])
end
```

当梁的一端固定 $u = 0$，另一端自由 $u = 1$，且 $M_0/m_0 = 0.2$ 时，执行上述程序。首先在MATLAB命令窗口中显示下述选择，然后产生图9.19和图9.20：

```
Enter one of the following numbers to select the bc at x = 0:
 [1] clamped(w = dw/dx = 0)
 [2] simply supported(w = d^2w/dx^2 = 0)
 [3] linear spring(d^2w/dx^2 = 0 and d^3w/dx^3 = kw)
 [4] torsion spring(d^2w/dx^2 = bdw/dx and w = 0)
 [5] linear spring and torsion spring (d^2w/dx^2 = bdw/dx and d^3w/dx^3 = kw)
Enter number for bc at x = 0:1
```

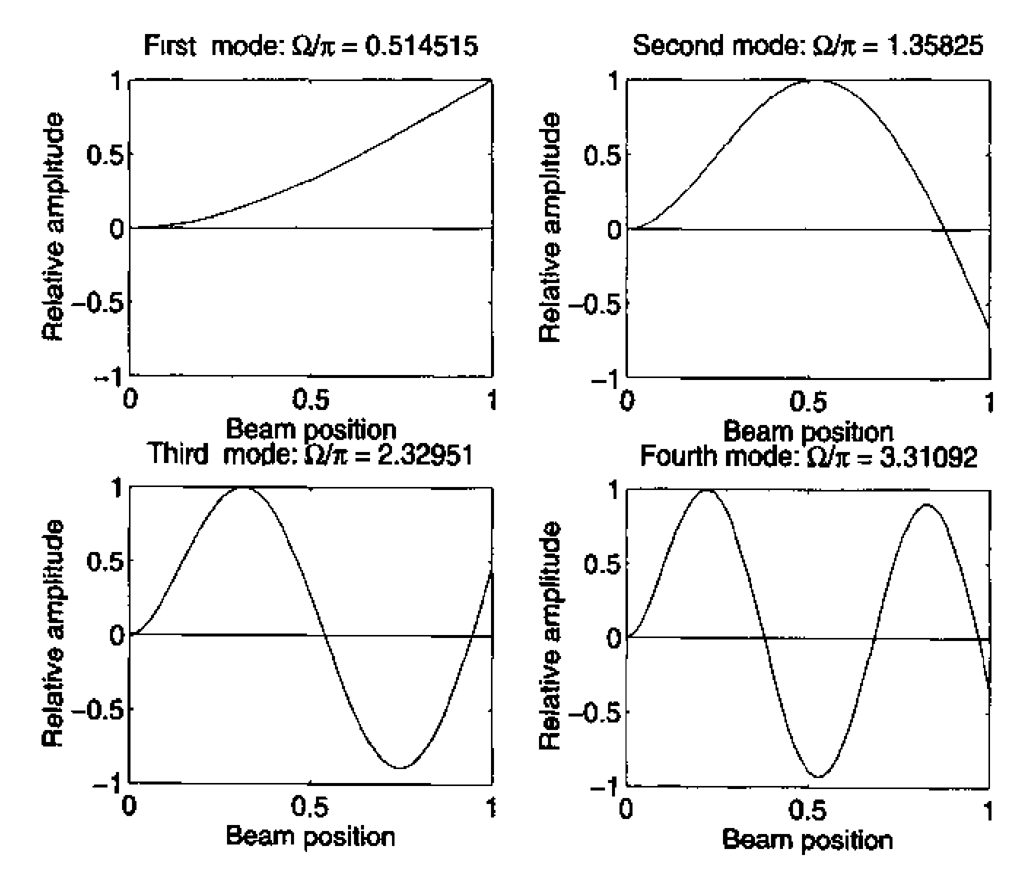


图 9.19 一端固定一端自由的梁在 $M_0/m_0 = 0.2$ 时得到的前四种模态

用户选择 1 后，接下来屏幕显示：

```
Enter one of the following numbers to select the bc at x = L:
 [1] clamped(w = dw/dx = 0)
 [2] simply supported(w = d^2w/dx^2 = 0)
 [3] vertically constrained slider with or without mass(dw/dx = 0 and d^3w/dx^3 = 0 or
     d^3w/dx^3 = mw)
 [4] linear spring with or without attached mass(d^2w/dx^2 = 0 and d^3w/dx^3 = 0 or d^
     3w/dx^3 = kw)
 [5] free with or without attached mass(d^2w/dx^2 = 0 and d^3w /dx^3 = 0 or d^3w/dx^3 =
     mw)
 [6] torsion spring(d^2w/dx^2 = bdw/dx and w = 0)
 [7] linear spring with or without attached mass and torsion spring (d^2w/dx^2 =
     bdw/dx and d^3w/dx^3 = kw)
Enter number for bc at x = L 5
```

用户选 5 后显示：

```
Enter ratio of attached mass to mass of beam (0 to 10000)
If no mass the enter zero.0.2
```

其中用户输入值为0.2。

还可运行下列程序得到图9.20所示结果。该程序中，τ 的变化范围从0到 $2\pi/\Omega_1$，且增量取0.05，此值在 $M_0/m_0=0.2$：$2\pi/\Omega 1\approx 3.9$ 时约等于其周期：

```
nfnum = 20; u = 0:0.02:1; xi = 0.4; M = 2;
tau = 0.1:0.05:3.9;
gg = GetBC;
[nffile, modefile] = BeamEqn(gg(7:10));
nfcoeff = ManyZeros(nffile, nfnum, 0.4, 1e-6, .4, gg(1:6));
for j = 1:nfnum
  wnx(j) = feval(modefile, nfcoeff(j), gg(1:6), xi);
  [mode, c] = feval(modefile, nfcoeff(j), gg(1:6), u);
  wnu(j, :) = mode;
  nn(j) = c;
end
wtau = (wnu'.* repmat(wnx./(nn.* nfcoeff.^2), length(u), 1)) * sin(nfcoeff * tau);
set(gca, 'nextplot', 'replacechildren')
for j = 1:length(tau)
  plot(u', -wtau(:, j))
  axis([0 1 -.2 .2])
  BeamFrame(j) = getframe;
end
movie(BeamFrame, M)
```

set命令用于阻止plot函数每次被调用时重新设置。getframe命令返回当前图形窗口的像素点。接着调用movie函数以 $M=2$ 次显示BeamFrame中所有像素点。

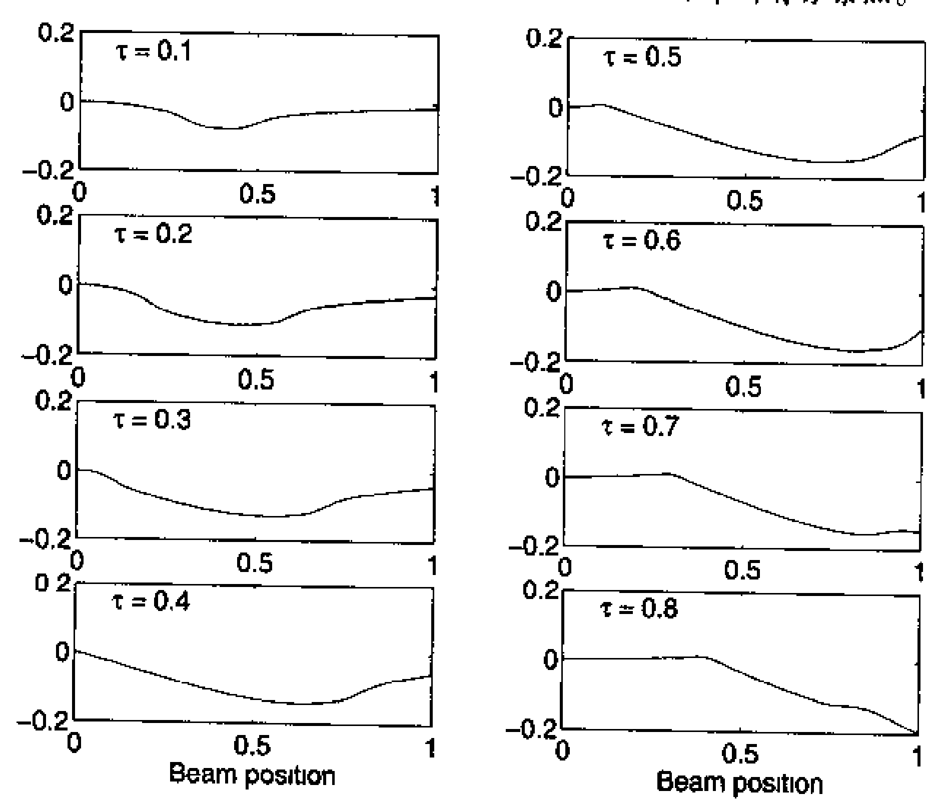


图9.20 当 $\zeta=0.4$ 的脉冲时一端固定一端自由的梁在 $M_0/m_0=0.2$ 时的响应曲线

9.4.2 变截面梁

图 9.21 所示一变截面梁，在 z 方向做简谐振动的无量纲振动方程为：

$$\frac{d^2}{du^2}\left[I(u)\frac{d^2 w}{du^2}\right]-\Omega'^4 A(u)w=0 \tag{9.33}$$

其中，

$$I(u)=[\alpha+(1-\alpha)u]^3[\beta+(1-\beta)u^{n-1}]$$
$$A(u)=[\alpha+(1-\alpha)u][\beta+(1-\beta)u^{n-1}]$$
$$\Omega'^4=\omega^2L^4/c_b^2r_0^2$$

r_0 为 $u=1$ 端截面的回转半径，$\alpha=h_1/h_0\leqslant 1$ 为锥深比，$\beta=b_1/b_0\leqslant 1$ 为锥的厚度比值，n 为大于 0 的常数，这种梁称做双锥面梁。$\beta=0$ 时梁的 $u=0$ 端在 xz 平面锥面缩为一点；$\alpha=0$ 时，在 $u=0$ 端锥面将沿 yz 平面缩小为一点；$\beta=\alpha=1$ 时得到一等截面梁。当锥的厚度比和深度比与 u 成线性比例关系时，$n=2$；当锥的厚度为常数时（$\beta=1$），$n=1$；当厚度比与深度比相等时，$\beta=\alpha$。

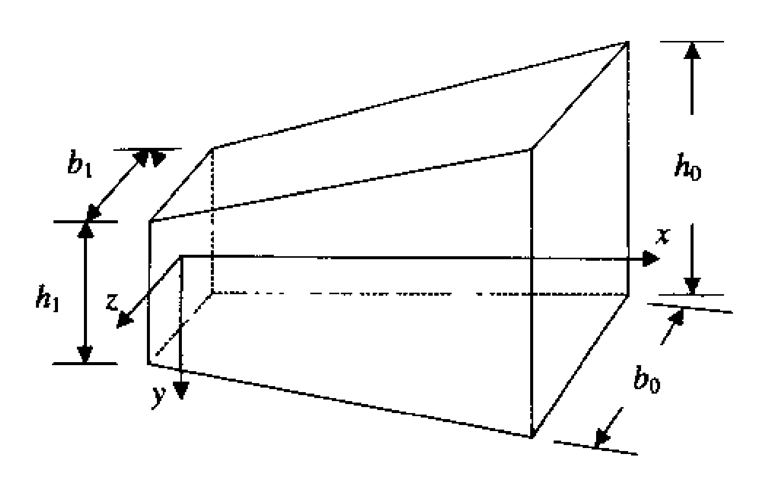


图 9.21 锥形梁的参数

下面缩小讨论范围，只讨论 $\beta=\alpha$ 及 $n=2$ 时的双锥面梁，则式(9.33)被写为：

$$\frac{d^2}{d\varphi^2}\left[\varphi^4\frac{d^2 w}{d\varphi^2}\right]-\lambda^4\varphi^2 w=0 \tag{9.34}$$

其中，

$$\varphi=[\alpha+(1-\alpha)u]$$
$$\lambda=\Omega'\alpha/(1-\alpha)$$

式(9. 34)的解为：

$$w(\varphi)=\varphi^{-1}[AJ_2(2\lambda\sqrt{\varphi})+BY_2(2\lambda\sqrt{\varphi})+CI_2(2\lambda\sqrt{\varphi})+DK_2(2\lambda\sqrt{\varphi})] \tag{9. 35}$$

其中 $J_2(z)$，$Y_2(z)$，$I_2(z)$，$K_2(z)$分别为一类和二类 2 阶贝塞尔函数及修改过的 2 阶贝塞尔函数。

对式(9.27a，b)用$[(1-\alpha)/\alpha]d/d\varphi$ 替换$\partial/\partial u$ 可求得边界条件。

例 9.2 双锥面悬臂梁的三个最低固有振动频率

假定一悬臂梁 $u=1$ 端固定，$u=0$ 端自由，现求其 α 在一段范围内取值时的三个最低固

有振动频率系数。边界条件为：

$u=1(\varphi=1)$：

$$w=0 \qquad \frac{\mathrm{d}w}{\mathrm{d}\varphi}=0 \tag{9.36A}$$

$u=0(\varphi=\alpha)$：

$$\frac{\mathrm{d}^2 w}{\mathrm{d}\varphi^2}=0 \qquad \frac{\mathrm{d}^3 w}{\mathrm{d}\varphi^3}=0 \tag{9.36B}$$

将式(9.35)代入式(9.36a,b)，得到下式，从而求得固有振动频率系数 λ_n。

$$\begin{vmatrix} J_5(2\lambda\sqrt{\alpha}) & Y_5(2\lambda\sqrt{\alpha}) & -I_5(2\lambda\sqrt{\alpha}) & K_5(2\lambda\sqrt{\alpha}) \\ J_4(2\lambda\sqrt{\alpha}) & Y_4(2\lambda\sqrt{\alpha}) & I_4(2\lambda\sqrt{\alpha}) & K_4(2\lambda\sqrt{\alpha}) \\ J_2(2\lambda) & Y_2(2\lambda) & I_2(2\lambda) & K_2(2\lambda) \\ J_3(2\lambda) & Y_3(2\lambda) & -I_3(2\lambda) & K_3(2\lambda) \end{vmatrix}=0 \tag{9.37}$$

为求得式(9.37)，用到下列等式：

$$\frac{\mathrm{d}}{\mathrm{d}\varphi}[\varphi^{-n/2}J_n(2\lambda\sqrt{\varphi})]=-\lambda\varphi^{-(n+1)/2}J_{n+1}(2\lambda\sqrt{\varphi})$$

$$\frac{\mathrm{d}}{\mathrm{d}\varphi}[\varphi^{-n/2}Y_n(2\lambda\sqrt{\varphi})]=-\lambda\varphi^{-(n+1)/2}Y_{n+1}(2\lambda\sqrt{\varphi})$$

$$\frac{\mathrm{d}}{\mathrm{d}\varphi}[\varphi^{-n/2}I_n(2\lambda\sqrt{\varphi})]=\lambda\varphi^{-(n+1)/2}I_{n+1}(2\lambda\sqrt{\varphi})$$

$$\frac{\mathrm{d}}{\mathrm{d}\varphi}[\varphi^{-n/2}K_n(2\lambda\sqrt{\varphi})]=-\lambda\varphi^{-(n+1)/2}K_{n+1}(2\lambda\sqrt{\varphi})$$

当 $\beta=\alpha=1$ 时，可通过执行例9.1中适用于悬臂梁情况的程序而求解，即表9.4中第4行标为 $\Lambda(\Omega n)$ 列的方程解。结果为 $\lambda_1=\Omega_1=1.8751$，$\lambda_2=\Omega_2=4.6941$，$\lambda_3=\Omega_3=7.8548$。

为求得式(9.37)的解，首先创建函数 TaperedBeam：

```
function r = TaperedBeam(x,alpha)
a1 = 2 * x;
a2 = a1 * sqrt(alpha);
r = det([besselj(5,a2)bessely(5,a2) - besseli(5,a2)besselk(5,a2);
        besselj(4,a2)bessely(4,a2)besseli(4,a2)besselk(4,a2);...
        besselj(2,a1)bessely(2,a1)besseli(2,a1)besselk(2,a1);...
        besselj(3,a1)bessely(3,a1) - besseli(3,a1)besselk(3,a1)]);
```

程序为：

```
c = [1 1.2 1.4 1.6 1.8 2 2.5 3 3.5 4 5 6 7 8 9 10];
a = 1./c;
for k = 2:length(a)
 b(k,:) = ManyZeros('TaperedBeam',3,0.4,1e - 6,0.4,a(k)) * (1 - a(k)/a(k);
end
b(1,:) = [1.8751 4.6941 7.8548];
semilogy(c,b,'k')
xlabel('1 / \alpha')
ylabel('Natural frequency coefficient( \lambda)')
```

```
title('Double-tapered cantilever beam')
text(8.5,20,'first')
text(8.5,32,'second')
text(8.5,45,'third')
```

其执行结果如图 9.22 所示。

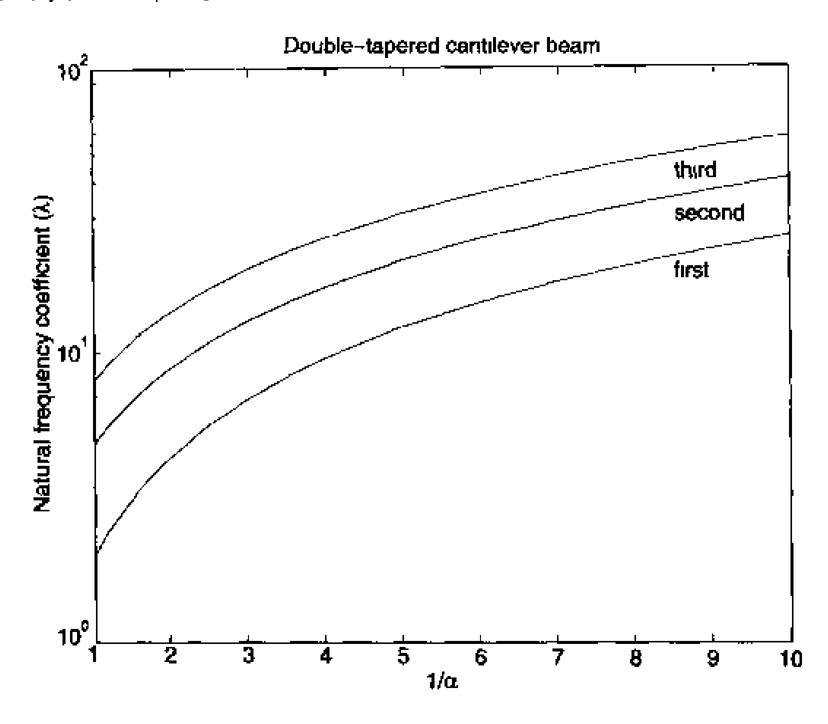


图 9.22 双锥面悬臂梁的前三个自然频率系数

9.4.3 有附加集中质量的梁

在 $u=\xi$ 处有附加集中质量 M'_0 的梁，做自由简谐振动的无量纲振动方程为：

$$\frac{d^4 w}{du^4} - \Omega^4\left[1+\frac{M'_0}{m_0}\delta(u-\xi)\right]w = 0 \tag{9.38}$$

其中 $w=w(u)$，边界条件由式(9.27a,b)给出。式(9.38)的一般解为：

$$w(u)/w(\xi) = AU(\Omega u) + BV(\Omega u) + CS(\Omega u) + DT(\Omega u) + \frac{M'_0}{m_0}\Omega T(\Omega(u-\xi))H(u-\xi) \tag{9.39}$$

其中 $H(u-\xi)$ 为单位阶跃函数：$H(u-\xi<0)=0$ 且 $H(u-\xi>0)=1$。频率系数 Ωn 由式(9.39)求出，取估计值 $u=\xi$，因 $T(0)=0$，所以有：

$$1 = AU(\Omega_n\xi) + BV((\Omega_n\xi)) + CS(\Omega_n\xi) + DT(\Omega_n\xi) \tag{9.40}$$

将式(9.27a,b)的边界条件代入式(9.39)求得常数 $A,\cdots,D$ 的值。对应的模态为：

$$w_n(u) = A_nU(\Omega_n u) + B_nV(\Omega_u u) + C_nS(\Omega_u u) + D_nT(\Omega_u u) + \frac{M'_0}{m_0}\Omega_n T(\Omega_n(u-\xi))H(u-\xi) \tag{9.41}$$

下面用例 9.3 说明这些结果。

例9.3 考虑带质量简支梁的模态

一两端简支梁，其质量所在位置ξ从0.05变到0.5，M_0/m_0有三种值0.1，1，10。下面求解梁的前三个固有频率系数，并画出图形。由于梁两端边界条件相同，所以Ω_n关于$\xi=0.5$对称。另外，在图中画出了各种模态，并将它们与两端简单支撑无附加质量的梁的模态进行了比较。

首先，边界条件为：

$$w(0)=\frac{\mathrm{d}^2w(0)}{\mathrm{d}u^2}=0$$
$$w(1)=\frac{\mathrm{d}^2w(1)}{\mathrm{d}u^2}=0 \tag{9.42}$$

将式(9.39)代入式(9.42)，发现$A=C=0$，且有：

$$B=\frac{M'_0\Omega}{G_0m_0}[T(\Omega)V(\Omega\varepsilon)-V(\Omega)T(\Omega\varepsilon)]$$
$$D=\frac{M'_0\Omega}{G_0m_0}[T(\Omega)T(\Omega\varepsilon)-V(\Omega)V(\Omega\varepsilon)] \tag{9.43}$$
$$G_0=V^2(\Omega)-T^2(\Omega)$$

其中$\varepsilon=1-\xi$，则式(9.40)变为：

$$V^2(\Omega_n)-T^2(\Omega_n)-\frac{M'_0}{m_0}\Omega_n\{V(\Omega_n\xi)[T(\Omega_n)V(\Omega_n\varepsilon)-V(\Omega_n)T(\Omega_n\varepsilon)]$$
$$+T(\Omega_n\xi)[T(\Omega_n)T(\Omega_n\varepsilon)-V(\Omega_n)V(\Omega_n\varepsilon)]\}=0 \tag{9.44}$$

在式(9.41)中引用式(9.44)，得到的模态为：

$$w_n(u)=B_nV(\Omega_nu)+D_nT(\Omega_nu)+\frac{M'_0}{m_0}\Omega_nT(\Omega_n(u-\xi))H(u-\xi) \tag{9.45}$$

其中B_n与D_n在$\Omega=\Omega_n$时由式(9.43)给出。

对无附加质量的两端简支梁，采用表9.4第一行的边界条件，发现$\Omega_n=n\pi$且$w_n(u)=\sin(n\pi u)$

首先创建下列函数以表示式(9.44)：

```
funtion nf = BeamMass(om,v)
ee = 1 - v(1);
p1 = Vvib(om)^2 - Tvib(om)^2;
p2 = Vvib(om * v(1)) * (Tvib(om) * Vvib(om * ee) - Vvib(om) * Tvib(om * ee));
p3 = Tvib(om * v(1)) * (Tvib(om) * Tvib(om * ee) - Vvib(om) * Vvib(om * ee));
nf = p1 - v(2) * om * (p2 + p3);
```

其中$v(1)=\xi$，$v(2)=M'_0/m_0$，*Vvib*和*Tvib*函数在本章末尾附录B中给出。

接下来创建一函数用于计算式(9.45)的模态：

```
function[modeshape,uu] = BeamMassMode(om,xi)
ee = 1 - xi;
p1 = Vvib(om)^2 - Tvib(om)^2;
p2 = (Tvib(om) * Vvib(om * ee) - Vvib(om) * Tvib(om * ee))/p1;
p3 = (Tvib(om) * Tvib(om * ee) - Vvib(om) * Vvib(om * ee))/p1;
```

```
ul = 0:0.01:xi;
modeshape1 = Vvib(om * ul) * p2 + Tvib(om * ul) * p3;
u = xi + 0.01:0.01:1;
modeshape2 = Vvib(om * u) * p2 + Tvib(om * u) * p3 + Tvib(om * (u - xi));
modeshape = [modeshape1 modeshape2];
modeshape = modeshape/max(abs(modeshape));
uu = [ul u];
```

总程序为:

```
mo = [.1 1 10];
xi = linspace(0.05,.5,10);
form = 1:length(mo)
 for k = 1:length(xi)
  coeff(m,k,:) = ManyZeros('BeamMass',3,0.4,le-6,0.4,[xi(k),mo(m)])/pi;
 end
end
lab = ['first';'second';'third'];
figure(1)
for kk = 1:3
 plot(xi,coeff(1,:,kk),'k-',xi,coeff(2,:,kk),'k--',xi,coeff(3,:,kk),'k:')
 hold on
 plot([0.05 0.1],[kk kk + .2],'k-')
 text(0.1,kk + .2,'M/m = 0')
 text(.4,kk + .1,lab(kk,:))
end
axis([0.05 .5 0 3.5])
xlabel('Mass position')
ylabel('\Omega \Lambda pi')
title('Simply supportd beam with attached mass')
legend(['M/m = 'num2str(mo(1))],['M/m = 'num2str(mo(2))],...
        ['M/m = 'num2str(mo(3))],3)
figure(2)
sig = 1;
lab = ['First mode';'Second mode';'Thoird mode'];
for k = 1:3
 if k == 3,sig = -1;end;
 for kk = 1:3
  subplot(3,3,3 * (k - 1) + kk)
  [shape,u] = BeamMassMode(coeff(kk,2,k) * pi,xi(8));
  plot(u,shape,'k-',u,sig * sin(k * u * pi),'k--',[0 1],[0 0],'k-')
  axis([0 1 -1 1])
  if k == 1,title(['M/m = 'num2str(mo(kk))],end;
if kk == 1,ylabel(lab < k,:)),end;
 end
end
```

执行结果如图 9.23 和图 9.24 所示。

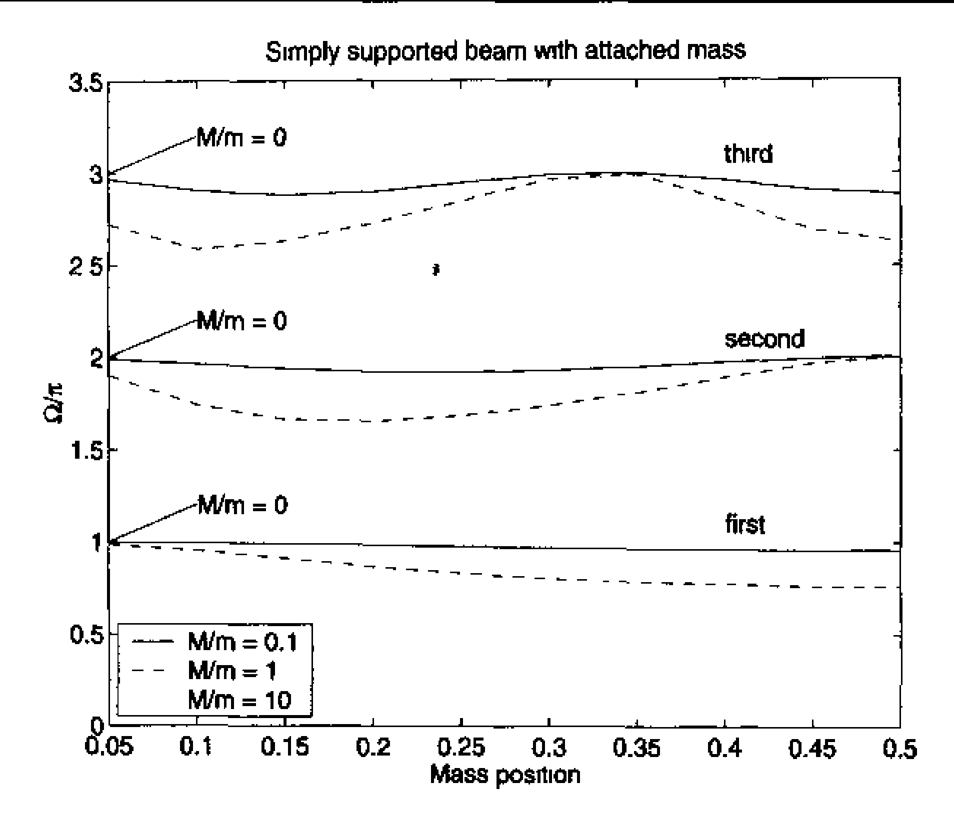


图 9.23　两端简支梁的固有频率系数随附加质量值及附加位置而变化的曲线

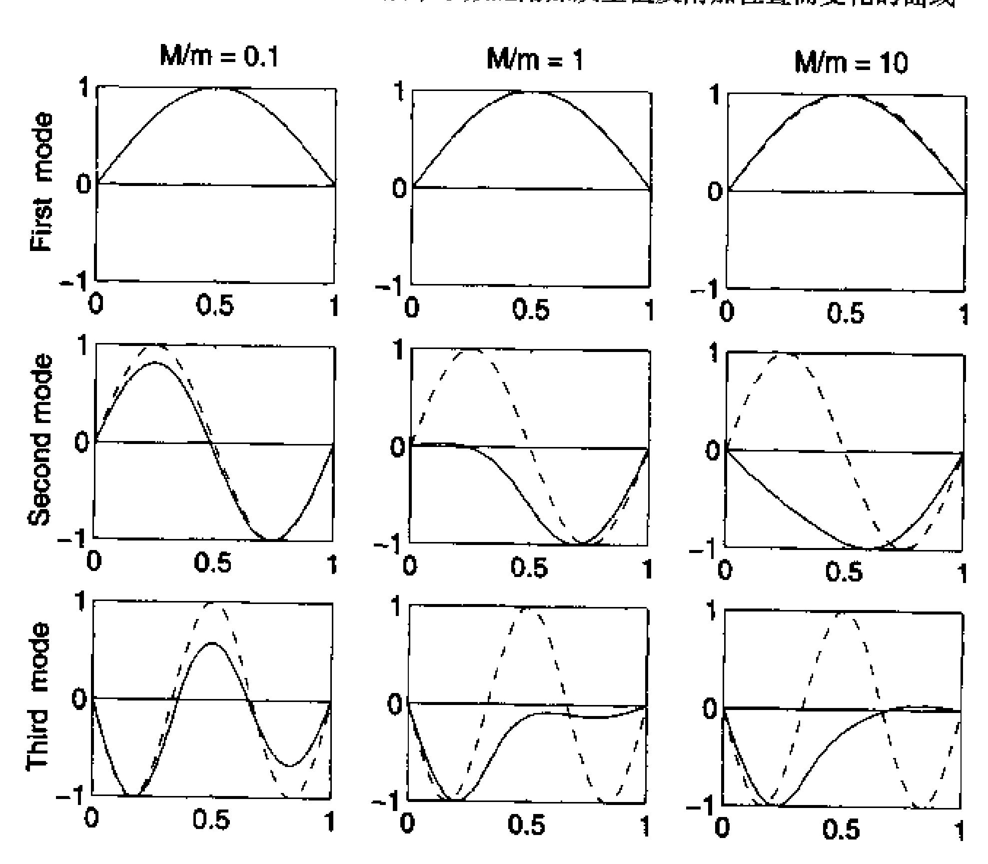


图 9.24　两端简支梁在 $\xi=0.4$ 处附加质量得到的模态：实线——带附加质量的梁；虚线——不带附加质量的梁

练习

9.1 由式(9.1)给出的两质量系统,在给定初始条件 $r(0)=2.0, \mathrm{d}r(0)/\mathrm{d}t=0.0, \theta(0)=0.0$ 及 $\mathrm{d}\theta(0)/\mathrm{d}t=0.5$ 时,用数值方法求其在(r,θ)平面上的轨迹。用 Ode45 函数求解时分别用到下列参数值:(a)每种状态下扫描时间取 20 单位,迭代步长为 20/400,相对误差为 10^{-3},绝对误差为 10^{-3};(b)每种状态下扫描时间取 20 单位,迭代步长为 20/400,相对误差为 10^{-6},绝对误差 10^{-6}。

判断每种情况下在 20 个扫描单位内单位质量的角动量 $r^2\theta$ 是否保持不变,并画出下列图形:(i)轨迹图,(ii)单位质量角动量对时间的关系图。

9.2 对式(9.8)的弹簧—质量—阻尼器系统的三种响应,用函数 *freeosc.m* 以数值方法求出欠阻尼系统、临界阻尼系统及过阻尼系统的响应,并将求得的响应结果与下述方程计算出的结果进行比较:①

欠阻尼系统:

$$x(\tau)=e^{-\zeta\tau}\left\{x_o\cos(\sqrt{1-\zeta^2}\,\tau)+\frac{v_o+\zeta x_o}{(\sqrt{1-\zeta^2}}\sin\sqrt{1-\zeta^2}\,\tau)\right\}$$

临界阻尼系统:

$$x(\tau)=[x_o+(v_o+x_o)\tau]e^{-\tau}$$

过阻尼系统:

$$x(\tau)=C_1e^{(-\zeta+\sqrt{\zeta^2-1})\tau}+C_2e^{(-\zeta-\sqrt{\zeta^2-1})\tau}$$

其中 $\tau=\omega_n t$,而且:

$$C_1=\frac{x_o(\zeta+\sqrt{\zeta^2-1})+v_o}{2\sqrt{\zeta^2-1}}$$

$$C_2=\frac{-x_o(\zeta-\sqrt{\zeta^2-1})-v_o}{2\sqrt{\zeta^2-1}}$$

9.3 一质量为 m 的物体沿长为 l 的粗糙杆滑动,杆绕 O 点旋转。杆在水平面的转动角度用 $\theta(t)$表示,物体沿杆滑动,以旋转点开始计算的滑动位置用 r 表示。当杆在水平面以固定角速度 $\mathrm{d}\theta/\mathrm{d}t=\omega$ 旋转时,m 的振动方程为:

$$r+2\mu\omega r-\omega^2r=0$$

其中 μ 为杆与物体间的摩擦系数,$\mu=0.2, l=3.0$ m, $\omega=6$ rad/s,初始条件为:$r(0)=1.0$ m,$r=0.0$ m/s,在图中画出物体离开杆前在(r,θ)平面内经过的路径。

9.4 加速计的测量值与真实幅值之比为:

$$\frac{A_m}{A_t}=\frac{1}{\sqrt{(1-r^2)^2+4\zeta^2r^2}}$$

其中 $r=\omega/\omega_n$, ω 为加速频率。ω_n 为加速计的固有频率,ζ 为加速计的阻尼因子。求:

① 参见 D J Inman,出处同上,第 1 3 节;S S Rao,出处同上,第 2 6 节。

(a) 画出 A_m/A_t 对 r 和 ζ 的关系图。

(b) 加速计的质量 $m=0.01$ kg，自然频率为 150 Hz，测量 6 000 r/m 加速度时误差 e 在下列值的 ±2.0% 范围内①，求加速计的阻尼因子：

$$e = 100\left(1-\frac{A_m}{A_t}\right)\%$$

9.5 一地震计测量值与真实幅值之比为②：

$$\frac{d_m}{d_t}=\frac{r^2}{\sqrt{(1-r^2)^2+4\zeta^2r^2}}$$

其中 $r=\omega/\omega_n$，ω 为加速频率；ω_n 为地震计的自然频率。ζ 为地震计的阻尼因子：

(a) 求 d_m/d_t 随 r 和 ζ 变化的曲线图。

(b) 求地震计在 500 r/m 到 1 500 r/m 测量范围内振动时，误差不超过 ±2.0% 的最大固有频率。

9.6 如图 9.25 所示的单自由度系统，在其平衡位置中心有一宽为 $2b$ 的死区。系统控制方程为：

$$m\ddot{x}+k(x+b)+2c\dot{x}=F(t)\qquad x<-b$$

$$m\ddot{x}+2c\dot{x}=F(t)\qquad -b\leqslant x\leqslant b$$

$$m\ddot{x}+k(x-b)+2c\dot{x}=F(t)\qquad x>b$$

当 $m=10.0$ kg，$k=150\times10^3$ N/m，$c=50$ Ns/m，初始条件为 $x(0)=0$，$\mathrm{d}x(0)/\mathrm{d}t=20$ m/s 时，求下列情况的自然响应：(a) 死区 $b=0.5$ μm，(b) $b=5.0$ μm，同时求出系统在 $F(t)=20\cos(12t)u(t)$ N 下的强迫响应，其中 $u(t)$ 为单位阶跃函数。

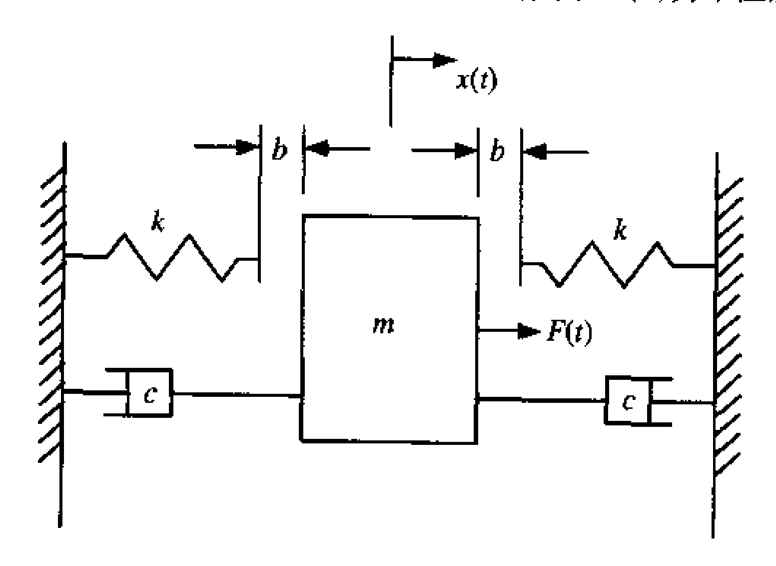


图 9.25　有死区的弹簧—质量—阻尼器系统

9.7 如图 9.26 所示的机械系统，在底部受到激励 $x_b(t)$ 时的传递函数为：

$$G(s)=\frac{x_m(s)}{x_b(s)}=\frac{2\zeta\omega_n s+\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$$

---

① 参见 D J Inman，出处同上，第 2 6 节；S S Rao，出处同上，第 10 3 节。

② 参见 S S Rao，出处同上，第 10.3 节。

用 bode 函数求出系统的振幅响应与相位响应，将所求结果与用下式分析的结果进行比较：

$$G(\omega)=\sqrt{\frac{\omega_n^4+(2\zeta\omega\omega_n)^2}{(\omega_n^2-\omega^2)^2+(2\zeta\omega\omega_n)^2}}$$

$$\phi(\omega)=\arctan\left(\frac{2\zeta\omega}{\omega_n}\right)-\arctan\left(\frac{2\zeta\omega\omega_n}{\omega_n^2-\omega^2}\right)$$

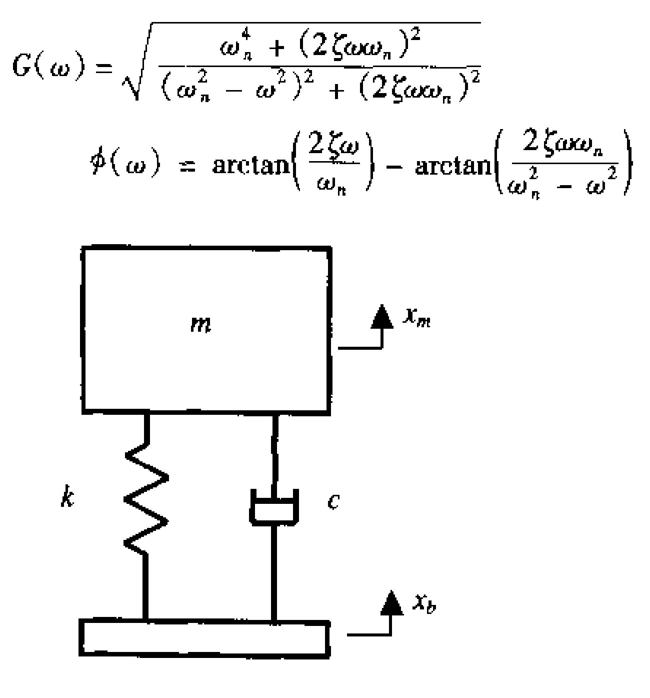


图 9.26　基部受激励的弹簧—质量—阻尼器系统

9.8 分析第 9.3.2 节讨论的减振器，在无阻尼的情况下，即 $c_1=c_2=0$，令第二级质量与主质量比值为 0.2，第二级质量的自然频率与主质量的自然频率比值为 1.0，画出 m_1 和 m_2 的无量纲频率响应关系图，并验证：(a)两图中谐振(极点)位置一致，(b) m_1 的响应曲线在减振器的自然频率处有一抗振点(零点)。

9.9 分析第 9.3.2 节讨论的减振器，令 $c_1=0$，第二级质量与主质量之比固定为 0.05，减振器的自然频率与主质量的自然频率之比为 1.0。减振器阻尼因子取不同值时，画出主质量的频率响应曲线图，并验证不同图形在两个频率处相交，如希望在前述两频率处响应振幅相等，则减振器自然频率与主质量自然频率之比应为多少？求出该减振器对应的 ξ 值，该减振器通常称为最优减振器。[①]

9.10 如图 9.27 所示的架空吊车，其小车通过钢索运输一质量为 m 的重物。当小车以加速度 $b(t)$移动时，吊车重物振动的控制方程为：

$$\frac{L\mathrm{d}^2\theta}{\mathrm{d}t^2}+g\sin\theta=-b(t)\cos\theta$$

其中 $g=9.8\ \mathrm{m/s^2}$ 为重力加速度。如果钢索长度为 2 m，在时间段，$0\leq t\leq 10$ s 内画出下列情况下的摇摆振动图 $\theta(t)$：

(a) $b(t)=5u(t)\ \mathrm{m/s^2}$，$\theta(0)=0.2$ rad 及 $\mathrm{d}\theta(0)/\mathrm{d}t=0$ rad/s。

(b) $b(t)=0.2u(t)\ \mathrm{m/s^2}$，$\theta(0)=0.2$ rad 及 $\mathrm{d}\theta(0)/\mathrm{d}t=0$ rad/s，其中 $u(t)$为单位阶跃函数。

① 参见 S S Rao，出处同上，第 9 10 节。

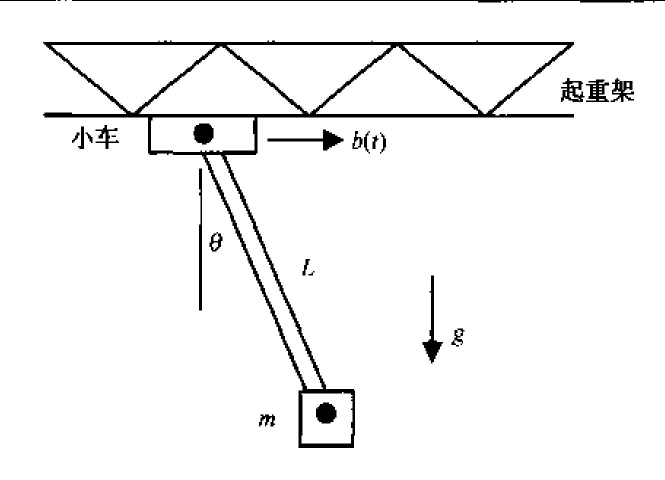


图 9.27　架空吊车上的小车运载一摇摆负载 m

9.11 图 9.28 所示的模型常用于分析车辆的弹性振动情况。该无阻尼系统的自由振动方程为：

$$m\frac{\mathrm{d}^2 x}{\mathrm{d}t^2}+(k_1+k_2)x+(L_2k_2-L_1k_1)\theta=0$$

$$I_c\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2}+(k_1L_1^2+k_2L_2^2)\theta+(L_2k_2-L_1k_1)x=0$$

如果 $k_1=1\,000$ lb/ft，$k_2=1\,500$ lb/ft，$L_1=5$ ft，$L_2=4$ ft，$m=50$ slug，$I_c=1\,000$ slug.ft^2，用特征值问题求出自然频率、模态及中心点位置。

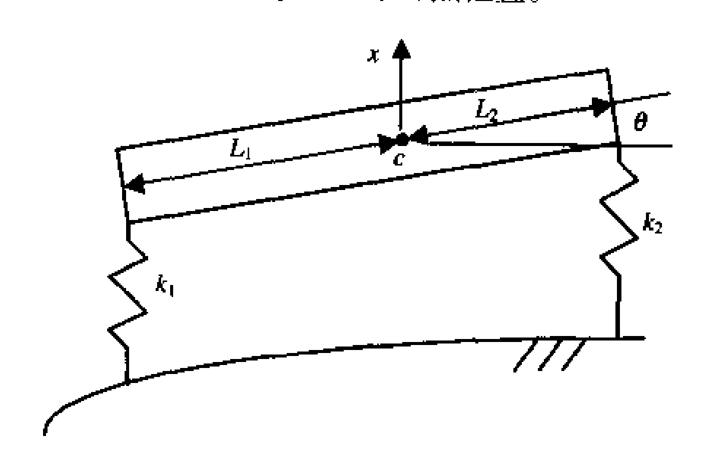


图 9.28　车辆的两自由度模型

9.12 车辆悬挂系统可用图 9.29 所示模型来表示。该系统的控制方程为：

$$m_1\frac{\mathrm{d}^2x_1}{\mathrm{d}t^2}+c_1\left(\frac{\mathrm{d}x_1}{\mathrm{d}t}-\frac{\mathrm{d}x_2}{\mathrm{d}t}\right)+k_1(x_1-x_2)=0$$

$$m_2\frac{\mathrm{d}^2x_2}{\mathrm{d}t^2}+c_1\left(\frac{\mathrm{d}x_2}{\mathrm{d}t}-\frac{\mathrm{d}x_1}{\mathrm{d}t}\right)+k_1(x_2-x_1)+k_2x_2+c_2x_2=k_2y+c_2\frac{\mathrm{d}y}{\mathrm{d}t}$$

在初始条件 $x_1(0)=0.5$ m，$\mathrm{d}x_1(0)/\mathrm{d}t=0$ m/s，$x_2(0)=0.2$ m，$\mathrm{d}x_2(0)\mathrm{d}t=0$ m/s 时，令 $m_1=1$ kg，$m_2=2$ kg，$c_1=1$ N/m/s，$c_2=5$ N/m/s，$k_1=10$ N/m，$k_2=30$ N/m，求系统的固有响应。对振动方程做拉普拉斯变换，并假定初始条件为 0，画出系统的频率响应曲线图。

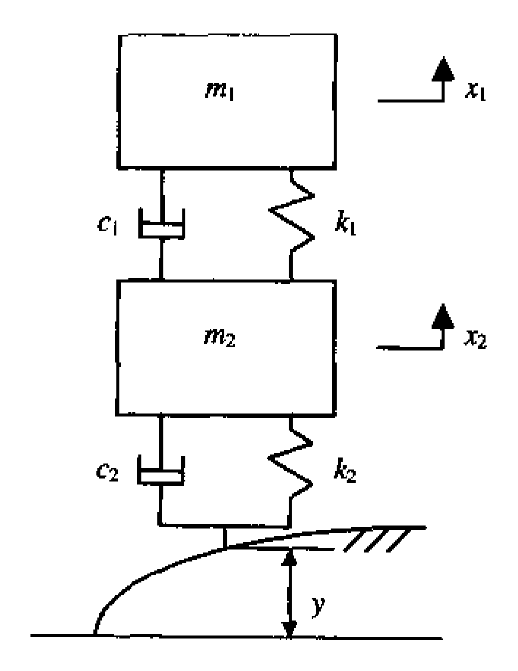


图 9.29 车辆悬挂系统的四分之一模型

9.13 如图 9.30 所示一循环转动刚圆轴两端受弹簧支撑，转子绕轴转动的角速度为 ω rad/s，转子沿转动轴方向有极惯性矩 J_p，转动平面内任何方向的横向惯性矩 J_t，各自水平方向的支撑刚性为 k_1 和 k_2。通过求解下述特征值问题可求得自由转动速度 Ω：

$$K^* w = \lambda M^* w$$

其中，

$$K^* = \begin{bmatrix} KM^{-1}K & KM^{-1}G \\ G'M^{-1}K & K + G'M^{-1}G \end{bmatrix} \quad M^* = \begin{bmatrix} K & 0 \\ 0 & M \end{bmatrix}$$

及

$$M = \begin{bmatrix} m & 0 & 0 & 0 \\ 0 & J_t & 0 & 0 \\ 0 & 0 & m & 0 \\ 0 & 0 & 0 & J_t \end{bmatrix} \quad G = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -J_p\omega \\ 0 & 0 & 0 & 0 \\ 0 & J_p\omega & 0 & 0 \end{bmatrix} \quad K = \begin{bmatrix} k_1 & 0 & 0 & 0 \\ 0 & k_2 & 0 & 0 \\ 0 & 0 & k_1 & 0 \\ & 0 & 0 & k_2 \end{bmatrix}$$

式中 $\lambda = \Omega^2$。

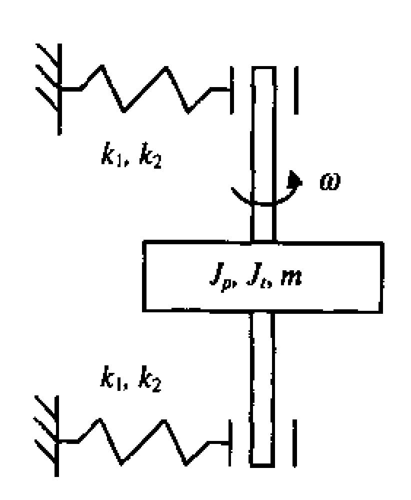


图 9.30 在弹性力支撑下的刚性转动转子

设 m = 10 kg, $J_p = 2\ \mathrm{kg\ m^2}$, $J_t = 1.2\ \mathrm{kg\ m^2}$, $k_1 = k_2 = 2.5\times10^6$ N/m,在图中画出随 Ω 变化的曲线,其中 $0\leqslant\omega\leqslant 1\,500$ rad/s。该图为坎贝尔图表① 的一个例子。在 $\Omega=\omega$ 处的速度称为临界速度,结果应如图 9.31 所示。

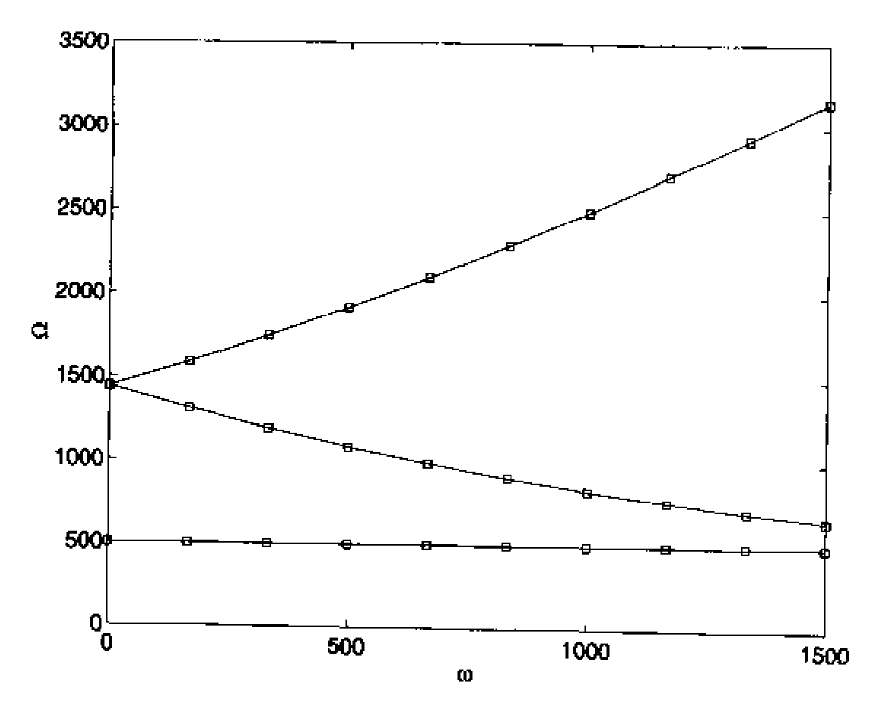


图 9.31 在弹性力支撑下刚性转子的坎贝尔图表

参考文献

1. Greenwood D T. Principles of Dynamics. 2nd ed. Prentice Hall, Englewood Cliffs, NJ, 1988.
2. Hale F J. Introduction to Space Flight, Prentice Hall, Englewood Cliffs, NJ, 1994.
3. Inman D J. Engineering Vibration. NJ: Prentice Hall, Englewood Cliffs, 1994.
4. Magrab E B. Vibrations of Elastic Structural Members, The Netherlands: Sijthoff & Noordhoff, 1979.
5. Meirovitch L. Elements of Vibration Analysis. New York: McGraw Hill, 1986.
6. Moon F C. Applied Dynamics with Applications to Multibody and Mechatronic Systems. New York: John Wiley & Sons, 1998.
7. Nayfeh A H. and Balachandran B. Applied Nonlinear Dynamics: Analytical, Computational, and Experimental Methods. New York: John Wiley & Sons, 1995.
8. Rao S S. Mechanical Vibrations. 3rd ed. MA: Addison-Wesley, Reading, 1995.
9. Tongue B H. Principles of Vibration. New York: Oxford University Press, 1996.

附录 B 第 9.4 节用到的 beam 函数

下列函数适用于计算在大范围变化的边界条件下等截面梁的自然频率及其模态。在例 9.1 中用到这些函数,现用 lambda 函数估算式(9.31)的解:

---

① 参见 G Genta, *Vibration of structures and Machines: Practical Aspects*, Springer-Verlag, New York, 1993 年,第 4 3 节。

```
function GammaOmega = lambda(x,z1,z2,z3,z4,z5,z6)
if x>20;
 ch = 0;
 tt = 1;
else
 ch = 1/cosh(x);
 tt = tanh(x);
end
ct = cos(x)* tt;
st = sin(x) * tt;
GammaOmega = z1 * (ct + sin(x)) + z2 * (ct - sin(x)) - 2 * z3 * st + ...
            z4 * (cos(x) - ch) + z5 * (cos(x) + ch) + 2 * z6 * cos(x);
```

现在定义 16 个函数 beamxxxx,这些函数为 lambda 提供了输入,lambda 函数将用于求满足 Ω_n 的值 $\Lambda(\Omega_n)=0$。参考式(9.27a,b)。该函数用到下列定义:

$v(1)=\gamma_2$
$v(2)=\alpha_2$
$v(3)=\gamma_1$
$v(4)=\alpha_1$
$v(5)=M_0/m_0$
$x=\Omega$

函数命名方法如表 9.5 所示。

函数 beam0000:

```
function c = beam0000(x,v)
b2 = - v(1)/x;
a2 = v(2)/(x^3) - v(5) * x;
b1 = v(3)/x;
a1 = v(4)/(x^3);
z1 = b1 * b2 * (a1 + a2) + (b1 - b2);
z2 = a1 * a2 * (b1 - b2) - (a1 + a2);
z3 = a1 * a2 + b1 * b2;
z4 = 1 - a1 * a2 * b1 * b2;
z5 = a2 * b2 - a1 * b1;
z6 = a1 * b2 - a2 * b1;
c = lambda(x,z1,z2,z3,z4,z5,z6);
```

函数 beam0001:

```
function c = beam0001(x,v)
b2 = - v(1)/x;
a2 = v(2)/(x^3) - v(5) * x;
b1 = v(3)/x;
a1 = v(4)/(x^3);
z1 = b1 * b2;
z2 = a2 * (b1 - b2 - 1);
z3 = a2;
z4 = a1 * b1 * b2;
z5 = - b1;
z6 = b2;
```

```
c = lambda(x,z1,z3,z3,z4,z5,z6);
```

函数 beam0010:

```
function c = beam0010(x,v)
b2 = - v(1)/x;
a2 = v(2)/(x^3) - v(5) * x;
a1 = v(4)/(x^3);
z1 = b2 * (a1 + a2) + 1
z2 = a1 * a2;
z3 = b2;
z4 = - a1 * a2 * b2;
z5 = - a1;
z6 = - a2;
c = lambda(x,z1,z2,z3,z4,z5,z6);
```

函数 beam0011:

```
function c = beam0011(x,v)
b2 = - v(1)/x;
a2 = v(2)/(x^3) - v(5) * x;
z1 = b2;
z2 = a2;
z3 = 0;
z4 = - a2 * b2;
z5 = - 1;
z6 = 0;
c = lambda(x,z1,z2,z3,z4,z5,z6);
```

函数 beam0100:

```
function c = beam0100(x,v)
b2 = - v(1)/x;
b1 = v(3)/x;
a1 = v(4)/(x^3);
z1 = b1 * b2;
z2 = a1 * (b1 - b2) - 1;
z3 = a1;
z4 = - a1 * b1 * b2;
z5 = b2;
z6 = - b1;
c = lambda(x,z1,z2,z3,z4,z5,z6);
```

函数 beam0101:

```
function c = beam0101(x,v)
b2 = - v(1)/x;
b1 = v(3)/x;
z1 = 0;
z2 = b1 - b2;
z3 = 1;
z4 = - b1 * b2;
z5 = 0;
z6 = 0;
```

```
c = lambda(x,z1,z2,z3,z4,z5,z6)
```

函数 beam0110:

```
function c = beam0110(x,v)
b2 = -v(1)/x;
a1 = v(4)/(x^3);
z1 = b2;
z2 = a1;
z3 = 0;
z4 = -a1 * b2;
z5 = 0;
z6 = -1;
c = lambda(x,z1,z2,z3,z4,z5,z6);
```

函数 beam0111:

```
function c = beam0111(x,v)
b2 = -v(1)/x;
z1 = 0;
z2 = 1;
z3 = 0;
z4 = -b2;
z5 = 0;
z6 = 0;
c = lambda(x,z1,z2,z3,z4,z5,z6);
```

函数 beam1000:

```
function c = beam1000(x,v)
a2 = v(2)/(x^3) - v(5) * x;
b1 = v(3)/x;
a1 = v(4)/(x^3);
z1 = b1 * (a1 + a2) - 1;
z2 = -a1 * a2;
z3 = b1;
z4 = -a1 * a2 * b1;
z5 = a2;
z6 = a1;
c = lambda(x,z1,z2,z3,z4,z5,z6);
```

函数 beam1001:

```
function c = beam1001(x,v)
a2 = v(2)/(x^3) - v(5) * x;
b1 = v(3)/x;
z1 = b1;
z2 = -a2;
z3 = 0;
z4 = -a2 * b1;
z5 = 0;
z6 = 1;
c = lambda(x,z1,z2,z3,z4,z5,z6);
```

函数 beam1010:

```
function c = beam1010(x,v)
a2 = v(2)/(x^3) - v(5) * x;
a1 = v(4)/(x^3);
z1 = a1 + a2;
z2 = 0;
z3 = 1;
z4 = - a1 * a2;
z5 = 0;
z6 = 0;
c = lambda(x,z1,z2,z3,z4,z5,z6);
```

函数 beam1011:

```
function c = beam1011(x,v)
a2 = v(2)/(x^3) - v(5) * x;
z1 = 1;
z2 = 0;
z3 = 0;
z4 = - a2;
z5 = 0;
z6 = 0;
c = lambda(x,z1,z2,z3,z4,z5,z6);
```

函数 beam1100:

```
function c = beam1100(x,v)
b1 = v(3)/x;
a1 = v(4)/(x^3);
z1 = b1;
z2 = - a1;
z3 = 0;
z4 = - a1 * b1;
z5 = 1;
z6 = 0;
c = lambda(x,z1,z2,z3,z4,z5,z6);
```

函数 beam1101:

```
function c = beam1101(x,v)
b1 = v(3)/x;
z1 = 0;
z2 = - 1;
z3 = 0;
z4 = - b1;
z5 = 0;
z6 = 0;
c = lambda(x,z1,z2,z3,z4,z5,z6);
```

函数 beam1110:

```
function c = beam1110(x,v)
a1 = v(4)/(x^3);
```

```
z1 = 1;
z2 = 0;
z3 = 0;
z4 = -a1;
z5 = 0;
z6 = 0;
c = lambda(x,z1,z2,z3,z4,z5,z6);
```

函数 beam1111:

```
function c = beam1111(x,v)
z1 = 0;
z2 = 0;
z3 = 0;
z4 = -1;
z5 = 0;
z6 = 0;
c = lambda(x,z1,z2,z3,z4,z5,z6);
```

下面给出 8 个函数 mode0xxx,这些函数用于求式(9.29)中的 $W_n(\eta)$以及求式(9.30)中 N_n 的值。在这些函数中有 $E_n = E_n$, $Nn = N_n$。

函数 mode0000:

```
function[mode,c] = mode0000(x,v,eta)
a2 = v(2)/(x^3) - v(5) * x;
b1 = v(3)/x;
a1 = v(4)/(x^3);
En = (a2 * Vvib(x) + (a2 * b1 - 1) * Svib(x) - b1 * Tvib(x))/(Vvib(x) - (a1 + a2) * Uvib(x) + ...
    a2 * a1 * Tvib(x));
mode = En * (Uvib(x * eta) - a1 * Tvib(x * eta)) + Vvib(x * eta) + b1 * Svib(x * eta);
c = Nn(x,En,1,b1, - a1 * En,v(5));
```

函数 mode0001:

```
function[mode,c] = mode0001(x,v,eta)
a2 = v(2)/(x^3) - v(5) * x;
b1 = v(3)/x;
En = (a2 * Vvib(x) + (a2 * b1 - 1) * Svib(x) - b1 * Tvib(x))/(Uvib(x) - a2 * Tvib(x));
mode = En * Tvib(x * eta) + Vvib(x * eta) + b1 * Svib(x * eta);
c = Nn(x,0,1,b1,En,v(5));
```

函数 mode0010:

```
function[mode,c] = mode0010(x,v,eta)
a2 = v(2)/(x^3) - v(5) * x;
a1 = v(4)/(x^3);
En = (a2 * Svib(x) - Tvib(x))/(Vvib(x)(a1 + a2) * Uvib(x) + a1 * a2 * Tvib(x));
mode = En * (Uvib(x * eta) - a1 * Tvib(x * eta)) + Svib(x * eta);
c = Nn(x,En,1,0, - a1 * En,v(5));
```

函数 mode0011:

```
function[mode,c] = mode0011(x,v,eta)
a2 = v(2)/(x^3) - v(5) * x;
```

```
En = (a2 * svib(x) - Tvib(x))/(Uvib(x) - a2 * Tvib(x));
mode = En * Tvib(x * eta) + Svib(x * eta);
c = Nn(x,0,0,1,En,v(5));
```

函数 mode0100:

```
function[mode,c] = mode0100(x,v,eta)
b1 = v(3)/x;
a1 = v(4)/(x^3);
En = (Vvib(x) + b1 * Svib(x))/( - Uvib(x) + a1 * Tvib(x));
mode = En * (Uvib(x * eta) - a1 * Tvib(x * eta)) + Vvib(x * eta) + b1 * Svib(x * eta);
c = Nn(x,En,1,b1,a1 * En,v(5));
```

函数 mode0101:

```
function[mode,c] = mode0101(x,v,eta)
b1 = v(3)/x;
En = (Vvib(x) + b1 * Svib(x))/( - Tvib(x));
mode = En * Tvib(x * eta) + Vvib(x * eta) + b1 * Svib(x * eta);
c = Nn(x,0,1,b1,En,v(5));
```

函数 mode0110:

```
function[mode,c] = mode0110(x,v,eta)
a1 = v(4)/(x^3);
En = (Svib(x))/( - Uvib9x) + a1 * Tvib(x));
mode = En * (Uvib(x * eta) - a1 * Tvib(x * eta)) + Svib(x * eta);
c = Nn(x,En,0,1, - a1 * En,v(5));
```

函数 mode0111:

```
function[mode,c] = mode0111(x,v,eta)
En = - Svib(x)/Tvib(x);
mode = En * Tvib(x * eta) + Svib(x * eta);
c = Nn(x,0,0,1,En,v(5));
```

N_n 由函数 Nn 求得，函数代码如下：

```
function nn = Nn(x,a,b,c,d,mo)
c1 = x * (a^2 + c^2 - 2 * b * d) + c * b - 3 * a * d + 0.25 * ((a - c)^2 - (b - d)^2) * sin(2 * x) - ...
      0.5 * (b - d) * (a - c) * cos(2 * x);
c2 = 0.25 * ((b + d)^2 + (a + c)^2);
c3 = 0.5 * (b + d) * (a + c);
c4 = (a^2 - c^2 + d^2 - b^2) * cos(x);
c5 = (a^2 - c^2 - d^2 + b^2) * sin(x);
c6 = 2 * (a * b - c * d) * sin(x);
c7 = 2 * (a * d - c * b) * cos(x);
test1 = c2 * tanh(2 * x) + c3
test2 = (c4 + c6) * tanh(x) + c5 + c7;
nn = (0.25./x) * (c1 + cosh(2 * x) * test1 + cosh(x) * test2);
if mo ~ = 0
  nn = nn + mo * (a * Uvib(x) + b * Vvib(x) + c * Svib(x) + d. * Tvib(x))^2;
end
```

和

```
function ux = Uvib(x);
ux = 0.5 * (cos(x) - cosh(x));

function sx = Svib(x)
sx = 0.5 * (cosh(x) - cos(x));

function vx = Vvib(x)
vx = 0.5 * (sinh(x) + sin(x));

function tx = Tvib(x)
tx = 0.5 * (sinh(x) - sin(x));
```

梁两端的边界条件由函数 GetBc 给出提问，由用户输入选择值，GetBc 函数如下所述：

```
function g = GetBC
disp('Enter one of the following numbers to select the bc at x = 0:')
disp('[1]clamped(w = dw/dx = 0)')
disp('[2]simply supported (w = d^2w/dx^2 = 0)')
disp('[3]linear spring (d^2w/dx^2 = 0 and d^3w/dx^3 = kw)')
disp('[4]torsion spring (d^2w/dx^2 = bdw/dx and w = 0)')
disp('[5]linear spring and torsion spring (d^2w/dx^2 = bdw/dx and
          d^3w/dx^3 = kw)')
lhn = input('Enter number for bc at x = 0:')
alpha1 = 0; beta1 = 0;
switch lhn
   case 1
    g1 = -1; a1 = -1;
   case 2
    g1 = 0; a1 = -1;
   case 3
   alpha1 = input('Enter dimensionless linear spring constant (0.01 to 100,000):');
   g1 = 0; a1 = 0;
   case 4
    beta1 = input('Enter dimensionless torsion spring constant (0.01 to 100,000):') ;
      g1 = 0; a1 = -1;
   case 5
    alpha1 = input('Enter dimensionless linear spring constant (0.01 to 100,000):');
    beta1 = input('Enter dimensionless torsion spring constant (0.01 to 100,000):');
    g1 = 0; a1 = 0;
end
mo = 0; rhn = -1;
disp(' ')
disp('Enter one of the following numbers to select the bc at x = L:')
disp(' [1] clamped(w = dw/dx = 0)')
disp(' [2] simply supported(w = d^2w/dx^2 = 0)')
disp(' [3] vertically constrained slider with or without mass(dw/dx = 0 and...
           d^3w/dx^3 = 0')
disp('     or d^3w/dx^3 = mw')
disp(' [4] linear spring with or without attached mass(d^2w/dx^2 = 0 and...
           d^3w/dx^3 = kw)')
disp(' [5] free with or without attached mass (d^2w/dx^2 = 0 and d^3w/dx^3 = 0...
           or')
disp('     d^3w/dx^3 = mw)')
```

```
disp(' [6] torsion spring (d^2w/dx^2 = bdw/dx and w = 0)')
disp(' [7]linear spring with or without attached mass and torsion spring...
            d^2w/dx^2 = bdw/dx')
disp('      and d^3w/dx^3 = kw)')
rhn = input('Enter number for bc at x = L');
alpha2 = 0;beta2 = 0;
switch rhn
  case 1
   g2 = -1;a2 = -1;
  case 2
   g2 = 0;a2 = -1;
  case 3
   mo = -1;
   disp('Enter ratio of attached mass to mass of beam (0 to 10000).')
   mo = input(' If no mass then enter zero. ');
   g2 = -1;a2 = 0;
  case 4
   alpha2 = input('Enter dimensionless linear spring constant(0.01 to 100,000):');
   mo = -1;
   disp('Enter ratio of attached mass to mass of beam (0 to 10000)')
   mo = input(' If no mass then enter zero. ');
   g2 = 0;a2 = 0;
  case 5
   mo = -1;
   disp('Enter ratio of attached mass to mass of beam (0 to 10000).')
   mo = input(' If no mass then enter zero.');
   g2 = 0;a2 = 0;
  case 6
   beta2 = input('Enter dimensionless torsion spring constant (0.01 to 100,000):');
   g2 = 0;a2 = -1;
  case 7
   alpha2 = input('Enter dimensionless linear spring constant (0.01 to 100,000):');
   mo = -1;
   disp('Enter ratio of attached mass to mass of beam (0 to 10000).')
   mo = input(' If no mass then enter zero. ');
   beta2 = input('Enter dimensionless torsion spring constant (0.01 to 100,000):');
   g2 = 0;a2 = 0;
end
g = [beta2 alpha2 beta1 alpha1 mo 0 g2 a2 g1 a1];
```

在函数 BeamEqn 中会得到正确文件名，由该文件名可以找到估算的 $\Lambda(\Omega_n)$和 $W_n(u)$函数，如下所示：

```
function [nf,mo] = BeamEqn(g)
if g(1) == -1;d4 = '1';else;d4 = '0';end;
if g(2) == -1;d3 = '1';else;d3 = '0';end;
if g(3) == -1;d2 = '1';else;d2 = '0';end;
if g(4) == -1;d1 = '1';else;d1 = '0';end;
nf = ['beam' d4 d3 d2 d1];
mo = ['mode' '0' d3 d2 d1];
```

其中 $g(1)$，$g(2)$，$g(3)$，$g(4)$分别表示 b_2，a_2，b_1，a_1 的极限情况。

第10章 控制系统

本章介绍使用MATLAB控制工具箱和SIMULINK对控制系统进行描述、设计和计算。

10.1 控制系统设计概述

研究图10.1所示的控制系统可以发现，典型控制系统中一般包括用微分方程表示的设备模型(控制对象)，并具有一个或多个控制变量[输入 $u(t)$]和一个或多个输出 $y(t)$。对所有控制系统而言都有一个算法控制器，它接收操作命令[参考输入 $r(t)$]和对象输出，并计算控制变量。图10.1描述了串联控制系统，在此控制系统中控制器取得参考输入和输出之间的差，形成误差信号 $e(t)=r(t)-y(t)$，根据误差信号再产生对象输入 $u(t)$。控制器一般由计算机实现，但是在一些传统系统中还采用模拟电路或机械装置实现。

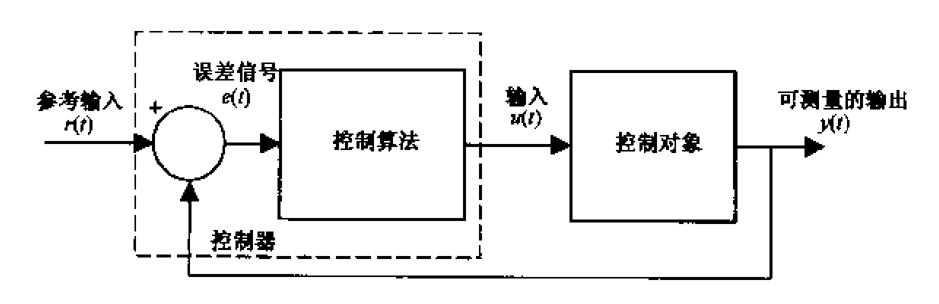


图10.1 反馈回路方框图

在控制系统中，首先应该明确系统的控制目标是什么。控制系统设计目标是改变物理系统使其符合要求。这不仅要求预知给定系统模型的变化，而且还要求能达到预期的目标。因此，控制系统设计存在着矛盾。

控制设计问题通常用反馈来说明。反馈的基本思想是用系统输出校正输入，使系统的运行不需人工干预。反馈系统除这个明显的优点外，还能减少非线性、增加系统的鲁棒性、提高系统的稳定性。反馈对控制工程师而言是非常有效(但不是惟一)的手段。

本章着重介绍如何使用MATLAB中的许多工具解决各种控制问题。第10.2节详细介绍在MATLAB中如何使用传递函数方法、方框图法及状态空间描述方法描述控制系统。第10.3节介绍控制系统的阶跃响应。第10.4节详细介绍波特图、根轨迹和LQR/LQG等设计方法。最后，第10.5节专门介绍设计实例。

10.1.1 控制器设计方法

控制器设计要求熟练掌握对描述实际系统特性的常微分方程的求解方法。然而在像MATLAB这样的计算工具出现以前，要想熟练掌握这一方法必须求解上百道微分方程问题。MATLAB中对线性时不变常微分方程有三种不同的描述方法：

1. 状态空间方程

2. 传递函数
3. 方框图

状态空间描述是在时域内用矩阵描述的,传递函数是在拉普拉斯域内使用复变量 s 的多项式描述的,在 MATLAB 中方框图描述是通过 SIMULINK 工具箱形像地描述输入和输出的连接。各种描述法之间的转换由 MATLAB 内置函数提供。控制设计目标通常分为三类:

1. 暂态
2. 稳态
3. 稳定性

暂态设计要求关注系统的短期行为,如敏感性和刚性。稳态设计要求关注系统的长期行为及一段时间内系统是如何运行的。标准输入信号有阶跃函数、斜坡函数和正弦函数,这些函数常被用来检验系统是否满足暂态及稳态设计的要求。MATLAB 中提供了可得到系统标准信号响应的函数。反馈对一个物理特性很好的系统而言,可能使系统稳定也可能使系统不稳定,而在实际中必须避免不稳定因素。暂态特性和稳态特性对设计要求通常是矛盾的,因此设计时应在两者之间进行折中。

用来解决控制问题的图解法包括:波特图、奈奎斯特(*Nyqust*)曲线和根轨迹,线性代数方法用于更先进的设计技术,如 LQG 方法、H_∞ 和 μ 综合方法等。对于大多数单入单出控制系统,使用一种控制器就能完成设计任务。

10.1.2 命名和文件约定

在本章中,使用了一套标准的命名约定。因为即使是一个简单的微分方程也需要多个向量和矩阵描述,所以 MATLAB 提供了一种方法,将必要的矩阵和向量集中到一个名称下。这种矩阵、向量甚至字符串的集合称为系统。*Plant* 用于标识系统名,在控制器设计过程中它的结构是不变的,而 *Control* 用于标识将要设计的部分。闭环系统由 *Plant* 及 *Control* 组成,用 *clSys* 标识。若断开反馈连接,则称系统为 *olSys*。本章中用来构成和分析控制系统的 MATLAB 函数把系统看做参数而不是看做向量和矩阵。

本章中研究几个系统实例,为方便起见,要创建用于返回系统模型的函数文件,函数返回系统对象。实例包括:

- 负载的永磁电机(*MotorSS.m*)
- 弹性轴驱动的指针(*Pointer.m*)
- 磁悬浮器(*MagLev.m*)
- 倒置摆(*Pend.m*)
- 飞轮(*Fly.m*)

根据需要可设计出超前、滞后、PI(proportional integral,比例积分)和 PD(proportional derivative,比例微分)等多种形式的控制器。

10.2 MATLAB 中系统的描述

控制系统的输入用 $r(t)$描述,这个函数一般用于描述控制中的某些变量,如力、电压或温

度。控制系统的输出用 $y(t)$描述,$r(t)$和 $y(t)$ 都是关于时间的实函数。函数值是一些可以测量的值,如角度、压力或速度。在控制系统中输入函数 $u(t)$和输出函数 $y(t)$之间的关系用线性时不变常微分方程表示,一般形式为:

$$a_n\frac{d^n y(t)}{dt^n}+a_{n-1}\frac{d^{n-1}y(t)}{dt^{n-1}}+\ldots+a_0 y(t)=b_m\frac{d^m u(t)}{dt^m}+b_{m-1}\frac{d^{m-1}u(t)}{dt^{m-1}}+\cdots+b_0 u(t) \quad (10.1)$$

其中 $n\geqslant m$。方程的系数 a_j 和 b_j 是定常实数。MATLAB 中把系统模型存储成对象,类似于第 6 章和第 7 章的图形对象,通过使用 get 和 set 函数很容易得到这些模型的特性。用式(10.1)所示的微分方程形式表示典型的控制问题,MATLAB 提供了三种输入—输出关系的类。

- 传递函数描述(类 tf)
- 状态空间描述(类 ss)
- 零点—极点—增益描述(类 zpk)

因为控制回路经常由计算机实现,所以离散线性系统在实际中用得很多。上述三种描述都有相应的离散时间形式,采样时间是附加的信息。把系统模型封装成对象,允许用户把相关数据追加到描述中。与追加数据相关的例子包括 InputName、OutputName 和 Notes。

为了举例说明三种基本的描述,研究图 10.2 所示的直流永磁伺服电机的简单模型。输入电压 $v(t)$作用于电动机的线圈上,产生电流 $i(t)$。线圈上的电流在转子上产生转矩,且与电流大小成比例。有效惯量 J 是负载惯量 J_1 和转子惯量 J_M 的和。具有角度 $\theta(t)$的转子就像发电机一样,产生与转子角速度成比例的反向电压。根据力矩平衡原理及电路分析原理可以得到下面两个线性常微分方程[①],用来描述输入电压 $v(t)$和输出角度 $\theta(t)$之间的关系:

$$L\frac{di(t)}{dt}+k_b\frac{d\theta(t)}{dt}+Ri(t)=v(t) \quad (10.2)$$

$$J\frac{d^2\theta(t)}{dt^2}+b\frac{d\theta(t)}{dt}-k_\tau i(t)=0$$

与电有关的常数是 R、L、k_τ 和 k_b,其中 R 是电动机的阻抗,L 是线圈的电感,k_τ 是从电流到转矩的转换因数,而 k_b 是发电机的反电动势常数。总惯量 J 通常取决于负载惯量 J_1。若没有变速箱,电机摩擦系数 b 一般很小。下面将这些方程用三种不同形式的模型表示。

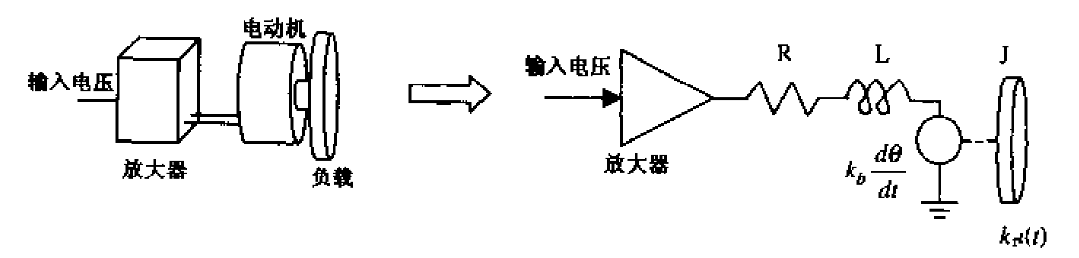


图 10.2 典型的伺服电机

10.2.1 状态空间模型

随着计算机的广泛应用,状态空间模型也得到了普遍的应用,因为它比传递函数更具有数

① 参见 D K.Anand 和 R B Zmood, *Introduction to Control Systems*,第三版,Butterworth - Heinemann Ltd, Oxford, England, 1995 年。

值稳定性。状态空间模型是相关的一阶微分方程组。因为存在 $\theta(t)$ 的二阶导数，所以图 10.2 所示的电动机模型不是一阶的。为了描述电机的状态空间模型，必须先把方程转化成第 5.5.4 节那样的一阶形式。设：

$$x_1(t)=\theta(t)$$

$$x_2(t)=\frac{\mathrm{d}\theta(t)}{\mathrm{d}t}=\omega(t)$$

$$x_3(t)=i(t)$$

则式(10.2)变为：

$$\frac{\mathrm{d}x_1}{\mathrm{d}t}=x_2$$

$$\frac{\mathrm{d}x_2}{\mathrm{d}t}=-\frac{b}{J}x_2+\frac{k_\tau}{J}x_3$$

$$\frac{\mathrm{d}x_3}{\mathrm{d}t}=-\frac{k_b}{L}x_2-\frac{R}{L}x_3+\frac{v(t)}{L}$$

如果令：

$$x(t)=[x_1(t),x_2(t),x_3(t)]'$$

$$u(t)=v(t)$$

$$y(t)=\theta(t)=x_1(t)$$

那么电机系统的状态空间描述为：

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t}=Ax(t)+Bu(t) \tag{10.3}$$

$$y(t)=Cx(t)+Du(t)$$

其中，

$$\boldsymbol{A}=\begin{bmatrix}0 & 1 & 0\\ 0 & -b/J & k_\tau/J\\ 0 & -k_b/L & -R/L\end{bmatrix}$$

$$\boldsymbol{B}=[0\quad 0\quad 1/L]'$$

$$\boldsymbol{C}=[1\quad 0\quad 0]$$

$$\boldsymbol{D}=[0]$$

在 MATLAB 中，矩阵 $\boldsymbol{A},\boldsymbol{B},\boldsymbol{C},\boldsymbol{D}$ 是描述微分方程的必要数据。

为方便地把模型输入到 MATLAB 中，可创建函数 MotorSS 来返回电机的状态空间系统模型。假设参数 $L=5$ mH(电机电感)，$R=5\,\Omega$(电机电阻)，$k_b=0.125$ V/rad/s(反电动势)，$k_\tau=15$ Nm/A(电机转矩常数)，$J=0.03\ \mathrm{kg\cdot m^2}$(电机惯性)和 $B=0.01$ Nm/rad/s(电机摩擦系数)，函数如下：

```
function Plant = MotorSS(J1)
if nargin < 1,J1 = 0;end;
L = 5e - 3;R = 5;kb = 12 5e - 2;
kt = 15;J = 3e - 2 + J1;b = 1e - 2;
A = [0 1 0;0 - b/J kt/J;0, - kb/L -R/L];
B = [0;0;1/L];
C = [1 0 0];
```

```
D=0;
Plant=ss(A,B,C,D);
set(Plant,'InputName','volts',OutputName','\theta');
set(Plant,'StateName',{'\theta','\omega','i'});
set(Plant,'Notes','Small DC servomotor');
```

函数 ss 把矩阵 $\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{C}$ 和 $\boldsymbol{D}$ 代入到单个系统对象中。键入

```
MotorSS
```

在 MATLAB 命令窗口中显示：

```
a =
              \theta    \omega      i
     \theta     0         1          0
     \omega     0      -0.33333     500
       i        0        -25       -1000
b =
              volts
     \theta     0
     \omega     0
       i       200
c =
              \theta    \omega      i
     \theta     1         0          0
d =
             volts
     \theta     0
Continuous-time model.
```

在默认情况下，输入用 $u1$ 表示，输出用 $y1$ 表示，内部状态变量为 $x1$、$x2$ 和 $x3$。可以用 set 函数提供这些标号，并将状态空间变量命名为 θ、ω 和 ι。此外，还要有注释说明模型所描述的内容。若没有对模型进行标志，则在 MATLAB 命令窗口中显示如下内容：

```
a =
          x1      x2        x3
     x1   0       1         0
     x2   0    -0.33333    500
     x3   0      -25      -1000
b =
          u1
     x1   0
     x2   0
     x3  200
c =
          x1      x2        x3
     y1   1       0         0
          u1
d =
     y1   0
Continuous-time system.
```

当该函数被调用时，只返回系统对象，而不返回常数。如果想重新得到矩阵 $\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{C}$ 和 $\boldsymbol{D}$，则可以使用：

```
[A,B,C,D] = ssdata(MotorSS)
```

返回系统矩阵 $\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{C}$、和 $\boldsymbol{D}$。

MATLAB 提供的内置函数可以把 *MotorSS* 这样的系统对象看做参数。将电机接到如图 10.3 所示的简单反馈系统中，如果要验证电机的特性，则框图中为：

$$v(t) = r(t) - y(t) = r(t) - \boldsymbol{C}x(t)$$

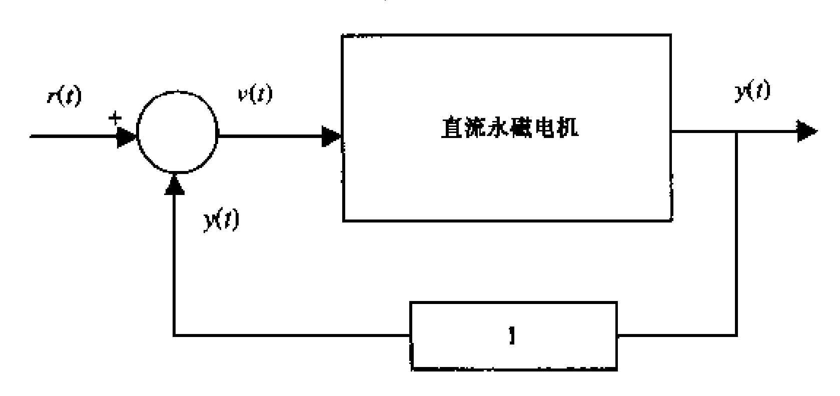


图 10.3 可控伺服电机的单位反馈控制系统

那么，式(10.3)相应地变成：

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = (\boldsymbol{A} - \boldsymbol{BC})x(t) + \boldsymbol{B}r(t)$$

$$y(t) = \boldsymbol{C}x(t) + \boldsymbol{D}r(t)$$

就是将式(10.3)中的 $\boldsymbol{A}$ 用 $\boldsymbol{A} - \boldsymbol{BC}$ 代替。MATLAB 提供的数学运算函数命令如下：

```
clSys = feedback(MotorSS,1);
```

它返回闭环系统。第二个参数中的数值 1 表示反馈回路的传递函数为 1。注意 *clSys* 中的 $\boldsymbol{A}$ 等于 *MotorSS* 中的 $\boldsymbol{A} - \boldsymbol{BC}$。可以通过输入下面的程序来检验：

```
clSys = feedback(MotorSS,1);
Plant = MotorSS;
clSys.a - (Plant.a - Plant.b * Plant.c)
```

返回(3×3)阶的零矩阵。

MATLAB 的函数 feedback 将算法与电机系统连成一个新的结构。在有输入时，将系统视为参数的 MATLAB 其他函数将用于求解微分方程。例如，下面的程序使用 step 函数计算阶跃输入信号的系统响应，曲线如图 10.4 所示。

```
[y,t] = step(feedback(MotorSS,1));
plot(t,y);
xlabel('Time');
ylabel('Rotor angle \ theta');
title('Step response of closed-loop servomotor');
```

下面的程序产生负载惯量变化时的伺服电机阶跃响应，曲线如图 10.5 所示。随着负载惯量 J_{load} 的增长，响应曲线出现了超调。在第 10.3 节中，根据函数零极点的位置可得出系统响应的期望值。

```
t=0:0.05:2;
```

```
Jload = 0:0.01:0.1;
data = zeros(length(t),length(Jload));
for i = 1:length(Jload)
 data(:,i) = step(feedback(MotorSS(Jload(i)),1),t);
end;
mesh(Jload,t,data);
view([45,30]);
xlabel('Load inertia J _ l');
ylabel('Time');
zlabel('Rotor angle \ theta(t)')
title('Step response of closed-loop servomotor')
```

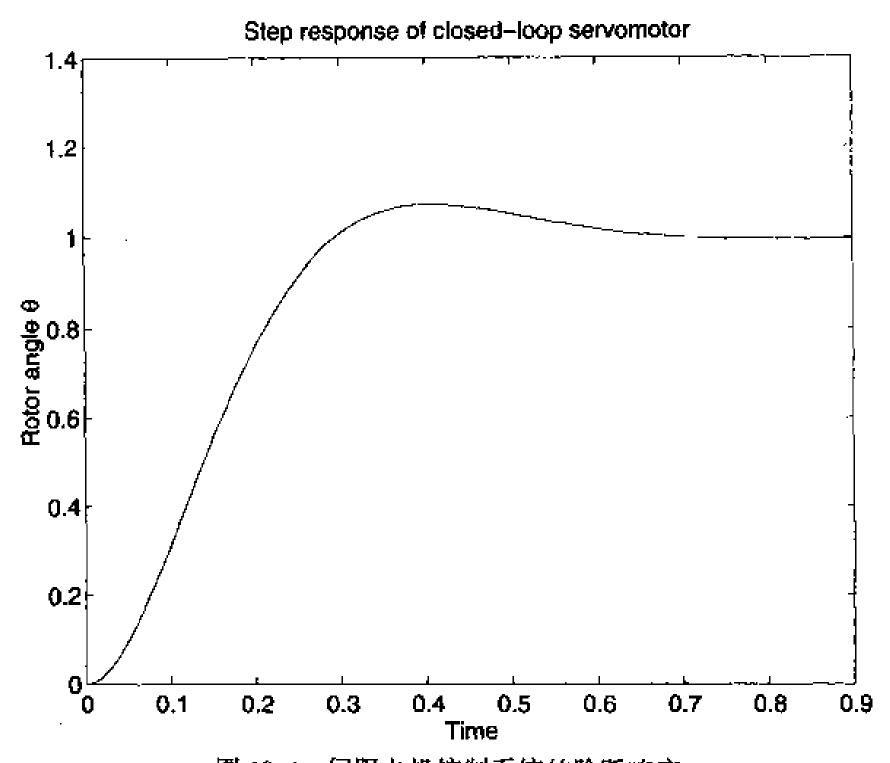


图 10.4　伺服电机控制系统的阶跃响应

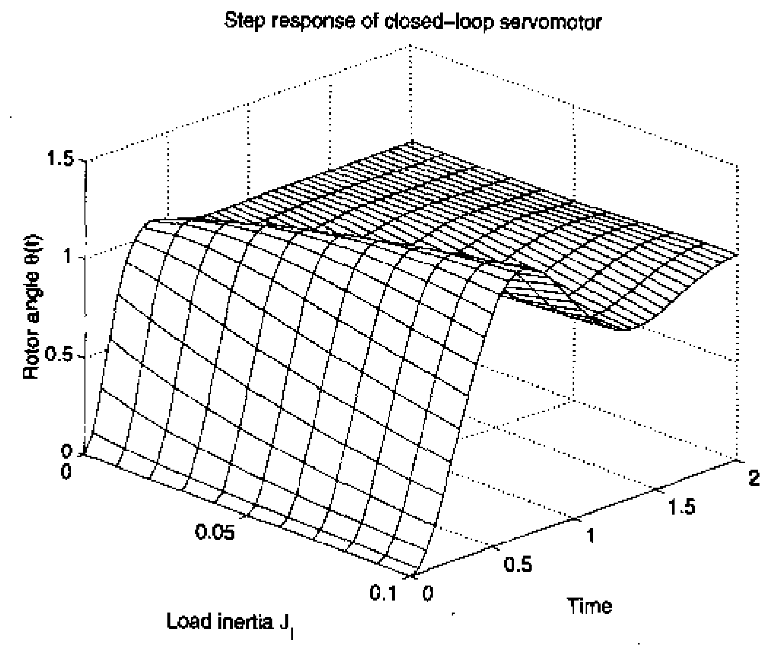


图 10.5　在不同负载惯量条件下闭环伺服电机的阶跃响应

10.2.2 传递函数描述

传递函数描述是指在零初始条件下,输出的拉氏变换除以输入的拉氏变换的结果。尽管这种描述不如状态空间方法应用普遍,而且对数值误差敏感,但是对设计者而言它仍然是非常直观的。[①] 式(10.1)所示的常微分方程的传递函数由下式给出:

$$H(s)=\frac{Y(s)}{R(s)}=\frac{b_m s^m+b_{m-1}s^{m-1}+\cdots+b_0}{a_n s^n+a_{n-1}s^{n-1}+\cdots+a_1 s+a_0} \tag{10.4}$$

其中 $a_n \neq 0$ 且 $n \geqslant m$,分母多项式的根称为系统的极点,分子多项式的根称为系统的零点。在 MATLAB 中用两个向量描述传递函数,每一个向量分别包含 s 的系数。分子多项式系数为:

```
Num=[bm...b1 b0]
```

分母多项式系数为:

```
Den=[an...a1 a0]
```

可以用 a_n 除以每个系数使 a_n 为 1,但这不是必须的。为了得到传递函数 $H(s)$,应对方程进行拉氏变换及必要的代数运算。对式(10.2)进行拉氏变换得到:

$$k_b s\Theta(s)+(sL-R)I(s)=V(s)$$

$$(Js^2+bs)\Theta(s)-k_\tau I(s)=0$$

假设初始条件为零,求解 $\Theta(s)$ 得:

$$\Theta(s)=\frac{k_\tau}{(sL+R)(Js^2+bs)+k_\tau k_b s} \tag{10.5}$$

系统传递函数模型可以由下面的程序产生,它将 *MotorSS* 产生的状态空间模型转换为传递函数模型:

```
function PlantTF=MotorTF
L=5e-3;R=5;kb=12.5e-2;
kt=15;J=3e-2;b=1e-2;
Num=kt;
Den=conv([L R],[J b 0])+[0 0 kb*kt 0];
PlantTF=tf(Num,Den);
```

键入下面的命令:

```
MotorTF
```

在 MATLAB 命令窗口中将会显示:

```
Transfer function:
              15
    
    0.00015 s^3+0.15s^2+1.925 s
```

分子多项式的系数 *Num* 和分母多项式的系数 *Den* 是传递函数得以实现的必要数据,正如矩阵

① 参见 *MATLAB Control System Toolbox, User's Guide*,第四修订版,The Math Works,Natick,MA,1992 年。

$\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{C}$ 和 $\boldsymbol{D}$ 是状态空间实现的必要数据一样。输入、输出和其他描述部分的名称可以通过使用 set 和 get 函数来设置。使用下面的命令可以从分子和分母多项式中获得系数：

```
[Num,Den] = tfdata(MotorTF,'v')
```

字符 v 说明函数将 *Num* 和 *Den* 以行向量形式返回，若省略 v，则分子将以元数组的形式返回。元数组主要用来描述 MIMO(multiple-input multiple-output，多入多出)系统。多数情况下，我们仅讨论 SISO(single-input single-output，单入单出)系统。与状态空间系统一样，MATLAB 函数也可以把传递函数作为输入参数。例如：

```
step(feedback(MotorTF,1))
```

将会计算并绘出控制电机的阶跃响应曲线。注意命令 step 和 feedback 与前面状态空间模型使用的命令相同，这些函数可在任何系统对象描述方法中运行。

另外两个有用的函数是 pole 和 tzero，函数 pole 返回极点(传递函数中分母多项式的根)。例如：

```
p = pole(feedback(MotorTF,1))
```

显示：

```
p =
1.0e+02*
 -9.8744
 -0.0645 + 0.0773i
 -0.0645 - 0.0773i
```

这是一个简单闭环控制系统的极点。函数 tzero 返回对象的传递零点：

```
z = tzero(feedback(MotorTF,1));
```

若 $\boldsymbol{Z}$ 为空矩阵，则说明系统没有零点。函数 pole 和 tzero 都可以用于状态空间系统。通过下式可实现传递函数与状态空间模型之间的转换：

$$H(s) = \boldsymbol{C}(s\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B}$$

其中 $\boldsymbol{I}$ 是单位矩阵。从方程可以看出，分母的根等价于行列式 $(s\boldsymbol{I} - \boldsymbol{A})$ 的根，也是矩阵 $\boldsymbol{A}$ 的特征值。极点、零点和特征值在分析控制系统特性中是经常用到的。

传递函数的一种描述形式是零点—极点—增益，它的参数是零点和极点。通常传递函数是一个有理多项式函数，所以其分子和分母表达式都可以分解成下面的形式：

$$H(s) = k\frac{(s - z_1)(s - z_2)\cdots(s - z_m)}{(s - p_1)(s - p_2)\cdots(s - p_n)} \tag{10.6}$$

传递函数由极点和零点及增益常数 k 惟一地确定。要生成如图(10.3)所示的闭环控制系统的零点—极点—增益模型，首先产生极点和零点，通过除以直流增益将增益设置为 1。程序为：

```
Poles = pole(feedback(MotorTF,1));
Z = tzero(feedback(MotorTF,1));
PlantZPK = zpk(Z,Poles,1);
PlantZPK = dcgain(feedback(MotorTF,1))/dcgain(PlantZPK) * PlantZPK;
step(PlantZPK);
```

程序执行后，同样产生如图10.4所示的曲线。命令 feedback、pole 和 tzero 也适用于这个模型。

10.2.3 离散时间模型

状态空间、传递函数和零点—极点—增益模型的离散时间形式可以用命令 c2d 产生。在下面的程序中，将连续时间模型转换为离散时间模型，采样时间为 1 ms。为了明显比较程序的图形输出，在连续时间系统响应与离散时间系统响应的结果之间加了一个很小的偏移，使得两条曲线分开。

```
[ydiscrete,time] = step(feedback(c2d(MotorSS,0.001),1));
ycontinous = step(feedback(MotorSS,1),time);
plot(time,ycontinous,'b-.',time,ydiscrete + 0.1,'b-');
grid;
xlabel('Time');
ylabel('Response');
legend('Continuous','Discrete',4)
title('Comparison of discrete-and continuous-time systems');
```

程序的结果如图10.6所示。由于采样时间很短，所以两条阶跃响应曲线之间没有明显差别。

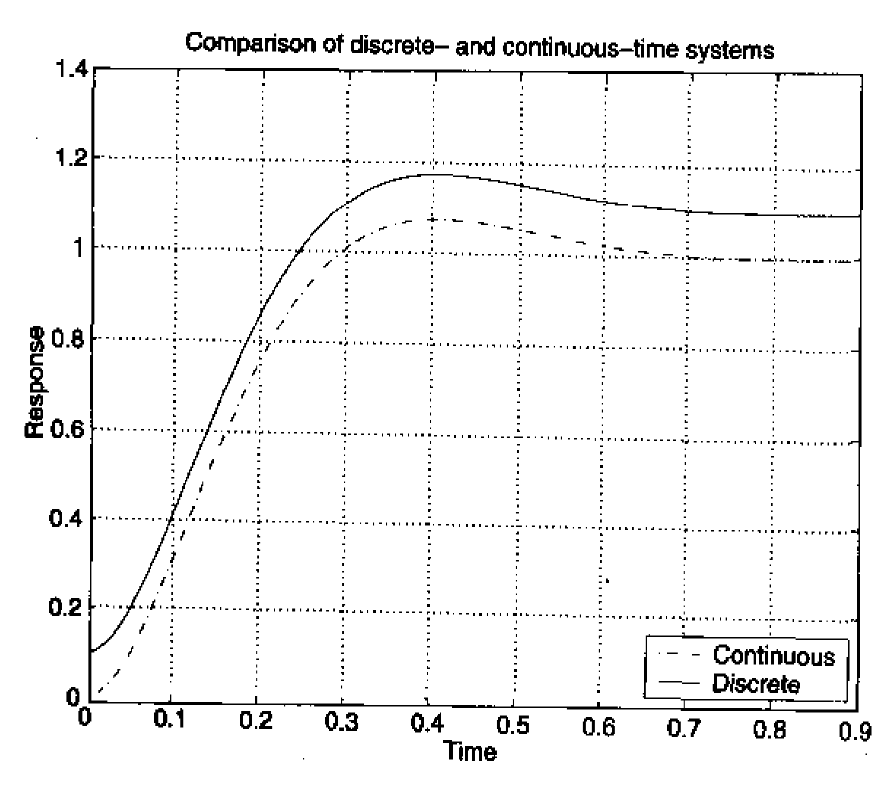


图10.6 离散时间阶跃响应与连续时间阶跃响应的比较（离散时间响应引入了0.1的垂直位移）

在默认情况下，函数 c2d 采用零阶保持器近似。函数 step、impulse 和 feedback 同样支持离散时间系统模型。当多个单元联合使用时，各个部分的采样时间应该相同。如果在 MATLAB 命令窗口键入：

```
c2d(MotorSS,0.001)
```

将显示下面的系统描述：

```
a =
              \theta      \omega        i
   \theta       1      0.00099818  0.00018374
   \omega       0        0.99507     0.31535
     i          0      -0.015768     0.36458
b =
              volts
   \theta  1.3203e-05
   \omega   0.036748
     i      0.12617
c =
              \theta      \omega        i
   \theta       1           0           0
d =
              volts
   \theta       0
```

注意离散时间模型的矩阵 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 与连续时间模型在本质上是不同的。离散时间模型的状态方程用差分方程描述，而不用微分方程。在状态空间模型中，用 k 表示时刻：

$$x[k+1]=\boldsymbol{A}_d x[k]+\boldsymbol{B}_d u[k]$$

$$y[k]=\boldsymbol{C}x[k]+\boldsymbol{D}u[k]$$

上式是矩阵的乘法运算。这些矩阵的乘法运算在采样时间 $t=k\Delta$ 与式(10.3)给出的连续时间微分方程的特征相匹配，其中 Δ 是适当选择矩阵 $\boldsymbol{A}_d$ 和 $\boldsymbol{B}_d$ 时的采样间隔，采用零阶保持器近似，为：①

$$\boldsymbol{A}_d=\mathrm{e}^{A\Delta}$$

$$\boldsymbol{B}_d=\int_0^{\Delta}\mathrm{e}^{A(\Delta-\tau)}\boldsymbol{B}\,\mathrm{d}\tau$$

将得到的左半个复平面的特征值映射到一个单位圆内。在 MATLAB 中还可得到三角近似、双线性近似(Tustin)、改进的双线性近似及匹配近似等多种方法。

10.2.4 方框图和仿真工具 SIMULINK

典型的控制系统由几个不同的单元组成，如对象单元和控制器单元等等。许多复杂结构的系统包括很多子系统，这些输入输出之间的连接图由系统设计者用方框图设计，基本元件组成如图 10.7 所示。MATLAB 提供了内置函数和运算符集合用于计算传递函数，其过程与方框图的代数运算类似。MATLAB 使用标准运算符，包括 +，-，*，/，并且还提供 feedback、series 和 connect 等函数，以便在命令行中执行和仿真方框图。用户可以利用 SIMULINK 工具箱提供的图形用户界面直接设计方框图，并且仿真特性曲线。

如图 10.7(a)所示，由乘法运算符确定两个系统的级联。例如，若系统对象 $H1$ 和 $H2$ 分别由传递函数 $H_1(s)$和 $H_2(s)$描述，那么级联系统 $H_3(s)$可以由下面的等式获得：

```
H3 = H1 * H2
```

① 参见 T Kailath, *Linear Systems Theory*, Prentice Hall, Englewood Cliffs, NJ, 1980 年；或 K Astrom 和 B. Wittenmark, *Computer Controlled Systems*, 第三版, Prentice Hall, Upper Saddle River, NJ, 1997 年。

除法也得到支持,但要求模型是严格正则的,在这里不进行讨论。

如图 10.7(b)所示,可以用加法和减法运算符得到两个系统的并联。若系统 $H1$ 和 $H2$ 并联,则最后的系统为:

```
H3 = H1 + H2
```

减法的情况可以通过改变上面等式的符号获得。

对于传递函数,图 10.7(c)所示的反馈连接不同于其他两种运算,执行过程不是用运算符实现而是用函数实现的。若 $H4$ 和 $H5$ 反馈连接,那么最后闭环系统为:

```
H3 = feedback(H4,H5);
```

以上假设为负反馈。若希望是正反馈,则 $H_3(s)$ 由下式得到:

```
H3 = feedback(H4 ,H5, +1);
```

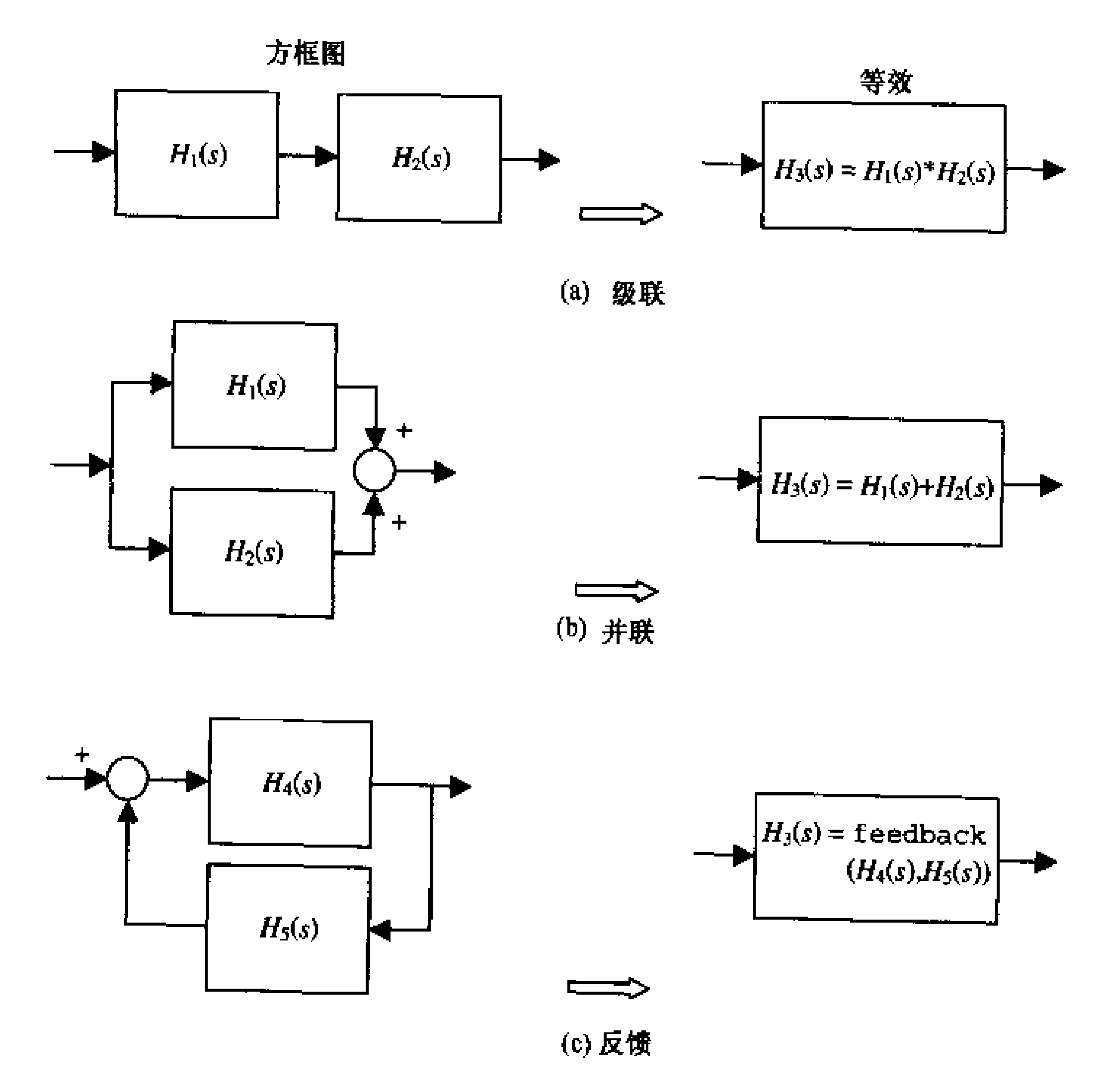


图 10.7 最常用的方框图代数运算

典型串联反馈控制结构如图 10.1 所示,它是由图 10.7(a)的串联形式和图 10.7(c)的反馈形式连接而成。在这种情况下,输入减去对象(H_2)的输出,形成误差信号反馈送给控制器(H_1)。下面建立闭环模型,设:

```
H4 = Controller * Plant
```

和

```
H5 = 1
```

则闭环系统 *clSys* 为：

```
clSys = feedback(Controller * Plant,1);
```

通过向量输入输出方式，函数 feedback 和 series 及操作符 *、+ 和 - 支持 MIMO 系统。此外，矩阵运算允许快速构建 MIMO 系统。例如，双入单出系统 *H*3，可由 SISO 系统 *H*1 和 *H*2 来创建，用下式实现：

```
H3 = [H1 ,H2]
```

单入双出系统可以这样来创建：

```
H3 = [H1;H2]
```

子系统 *H*1 和 *H*2 可以用第 2.4 节讨论的获得子矩阵的方法从 *H*3 中获取。在 MATLAB 系统中，串联、并联和反馈连接要求明确规定谁的输出连到谁的输入上去。函数 series 和 parallel 能够满足这个要求。许多复杂 MIMO 系统可以用 star 和 connect 来创建。在此仅研究 SISO 控制系统。

若系统对象 *H*1 和 *H*2 的类型不同，那么必须进行类型转换，以保证最后模型是统一的。在所有模型中，MATLAB 首先支持状态空间模型。因此，若计算过程中只要有一个模型为状态空间模型，那么结果就为状态空间形式。对于传递函数和零点—极点—增益模型而言，MATLAB 先支持传递函数模型。

SIMULINK SIMULINK 允许用户从元件库建模，并且对系统进行仿真。从元件库中选中元件并拖到模型窗口，连接各个不同的元件并设置每个元件参数，那么就可以运行模型并显示结果了。

SIMULINK 可以通过在 MATLAB 命令中键入下面的命令产生：

```
simulink
```

MATLAB 命令窗口中定义的变量可以在 SIMULINK 窗口中获得。

下面举例说明如何用 SIMULINK 建立直流伺服电机的模型，最后的结果如图 10.8 所示。为了产生对象模型，先在 MATLAB 命令窗口中定义并获得系统矩阵，然后打开 SIMULINK，即：

```
[A,B,C,D] = ssdata(MotorSS);
simulink
```

打开 SIMULINK 浏览窗口，显示四个主要的元件库：*Simulink*、*Control System Toolbox*、*Neural Network Blockset* 和 *Simulink Extras*。首先点击空白页标记打开 SIMULINK 模型窗口，然后再返回浏览窗口双击 SIMULINK，显示 SIMULINK 元件库的目录：

Continuous
Discrete
Functions & Tables

Math
Nonlinear
Signals & Systems
Sinks
Sources

若想打开任意库中的元件(模块),只须双击需要的元件库即可。先打开 Continuous 元件库。接着用鼠标单击 State - Space 模块,并按住键不放,把该元件拖到模型窗口。把元件放在期望的位置并放开鼠标。下一步,打开 Math 元件库,依次选中 Slider Gain 和 Sum 模块,并将其拖到模型窗口,都放在 State - Space 模块的左边。然后打开 Source 元件库选择 Step 模块,并将其放在模型窗口中所有模块的左边。最后从 Sinks 元件库选择 Scope 模块,将其放在所有模块的右边。

下一步要做的是设置模块的参数。从 State - Space 模块开始,双击该模块,将得到 State - space 模块参数窗口,有五个需要输入数据的地方:A、B、C、D 和 Initial condition。因为在 MATLAB 命令窗口中已经定义了这些矩阵,所以分别键入 A、B、C 和 D 即可。如果选择 MATLAB 命令窗口中定义的不同变量名,那么应该在相应的位置输入这些变量名。设初始状态为 0,可以采用同样的方式改变加号,第二个符号应改为负号。变增益模块的增益值是可以选择的,选择范围为 0 ~ 15,本例中选为 1。对 Step 模块,除 Step time(时间偏移或延迟)选为 0 外,其他为默认值。双击 Scope 模块,得到一个模拟的示波器,这时不需对其进行参数设置

连接模块形成反馈控制系统。设计窗口中的模块可以用鼠标移动,也可以修改其大小,每个模块外面的凸起部分都是用来连接的。默认情况下,输入在左边,输出在右边。利用鼠标从输出端拖到输入端即可连成一条线。将十字准线放在 Step 模块的输出位置,并按住鼠标左键不放,将十字准线移到 Sum 模块的加法(+)输入端。重复这种操作,直到将方框图连成为止,如图 10.8 所示。State - Space 模块和 Scope 模块中间一点到求和模块减法输入端的连线可以这样产生。将箭头放在两个模块之间的连线上,同时按下 Ctrl 键和鼠标左键,保持这种操作并将十字准线移到求和模块的减法输入端,放开鼠标和 Ctrl 键。若要调整连线,简单地单击连线并按住鼠标左键,即可将其移到需要的位置。

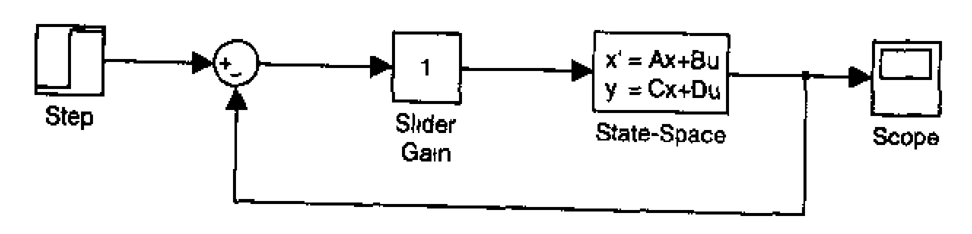


图 10.8 在 *SIMULINK* 中创建直流伺服电机框图

模块的位置和大小并不重要,关键是连接。在创建连接之前,需将一些模块进行反转,使得输入在左边,输出在右边。这种操作可以这样完成:先选中模块,然后在下拉菜单 Format 内选取 Filp Block 即可。另一种方式是在格式菜单中隐藏模块下面的注释,选中模块并选取 Hide Name 即可隐藏注释。若要再显示模块注释,选中模块并选取 Show Name 即可。但这两种选择不能同时出现。

若要进行仿真,在 Simulation 菜单中选取 Start,仿真结束后,双击 Scope 就可以看到仿真结果。用 x 轴和 y 轴放大图标可以调整图像清晰度。注意在 Simulation 菜单下,可以修改仿真参

数。例如,可以通过修改参数(选择 Parameters)改变综合方法。如果图形连接得不完全或不正确,则仿真不能进行,在 MATLAB 命令窗口会给出错误信息,并弹出一个特殊窗口。对图 10.8 所示的仿真模型在 Scope 中的结果如图 10.4 所示。

10.2.5 描述方法的转换

MATLAB 通过构造函数 ss、tf 和 zpk 进行三种描述之间的转换。例如,在 tf 中,用状态空间模型 MotorSS 产生传递函数模型。

```
PlantTF = tf(MotorSS)
```

转换过程中数值误差将产生微小的系统模型系数差异。

函数 ss 可以将传递函数转换为状态空间模型,因此:

```
PlantSS = ss(PlantTF)
```

转换后的系统矩阵 $\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{C}$ 和 $\boldsymbol{D}$ 与 *MotorSS* 定义的矩阵不同。这说明,对于给定的系统,其状态空间模型描述不是惟一的,这与传递函数模型不一样。一般使用:

```
ssbal
```

来调节输入量、状态量和输出量,使得仿真条件尽可能好。因此,用:

```
PlantBal = ssbal(PlantSS)
```

试图找到系统的最佳条件。

零点—极点—增益模型也可以由状态空间模型或传递函数模型转换。例如,零点—极点—增益模型可以由状态空间模型产生,用:

```
PlantZ = zpk(PlantSS)
```

这种转换与其他转换方法一样,在 MATLAB 中使用数值解有时会有很大的数值误差,尤其是对 10 阶以上的系统求解[①]。在实际情况中,建议不要经常变换描述方法。

10.3 系统响应

本节将介绍一些工具的用法,这些工具用来计算系统响应。三种系统描述使用的函数包括 step、impulse、initial 和 lsim。在使用方框图时仿真工具箱提供的许多内置信号源使仿真得到了简化。

无论是传递函数描述还是状态空间描述,配置零极点都是控制工程师估计系统响应的便捷方法,许多设计要求可以转化为零极点位置的约束条件。控制器的设计问题经常转变为反馈回路的设计,应使得闭环系统极点位于复平面中期望的位置,但是零点的位置不能通过反馈来改变。

---

① 参见 N.E.Leonard 和 W.S Levine, *Using MATLAB to Analyze and Design Control Systems*, Benjamin/Cummings, Redwood City, CA, 1992 年。

10.3.1 仿真

在控制系统的响应中，阶跃输入经常用来作为比较不同控制器的基准。MATLAB 函数 step 用于计算阶跃响应，若不需要返回值，则仅画出响应曲线。函数自动给出适当的时间范围来计算仿真，因此程序如下：

```
[theta,t] = step(MotorSS);
plot(t,theta)
title('Step response of a motor')
xlabel('Time')
ylabel('\theta(t)')
```

程序给出了数值并得到图 10.9。从图中可以看出，电动机的阶跃响应是一条斜线，因为恒定电压加在线圈上，所以电动机最后会达到恒定的转速。

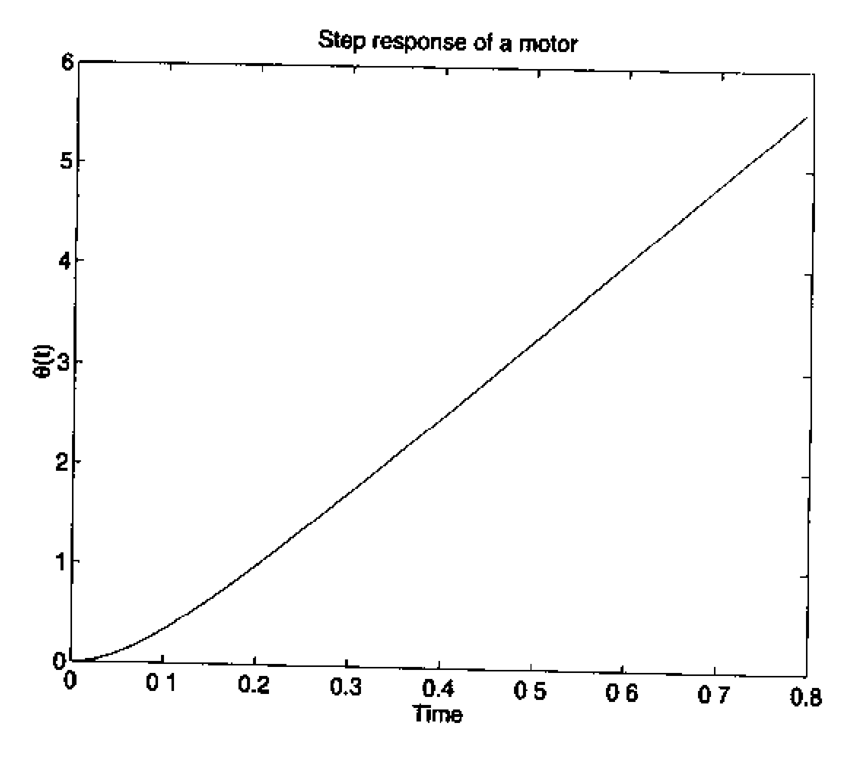


图 10.9 无反馈的电动机阶跃响应曲线

另一方面，可以不用 step 提供的时间范围而自己指定时间 t 的范围。在这种情况下，需将前面程序段的第一行改为下面的两行表述：

```
t = linspace(0,100,100);
theta = step(MotorSS,t);
```

其中 $0 \leqslant t \leqslant 100$。

MATLAB 函数 impulse 用于计算系统的脉冲响应，其用法与 step 类似。

MATLAB 提供的函数不能适应于所有的标准检测输入，但是使用下式则提供了在任意输入条件下求解系统响应的能力：

```
lsim(sys,u,t)
```

其中 sys 为研究中的系统，而 u 是描述输入幅值的向量，也是时间 t 的函数。u 和 t 的长度必须相等。

下面测定电动机控制系统跟踪误差的稳定特性。当输入是斜坡信号时，稳态误差是期望位置与实际位置之间的最终差值。在 $0 \leqslant t \leqslant 1$ 的范围内，输入信号斜率为 1，对直流伺服电机而言，当 $0 \leqslant t \leqslant 1$ 时，随动误差 $e(t)=\theta(t)-t$，程序为：

```
t = linspace(0,1,100);
theta = lsim(feedback(Motorss,1),t,t);
plot(t,t' - theta);
error = t(end) - theta(end);
hold on;
plot([0,t(end)],[error,error],'r--');
xlabel('Time');
ylabel('Error e(t)');
title('Error signal as a function of time')
```

执行后产生图 10.10。注意经过最初的暂态以后，误差近似稳定在 0.13。控制系统设计目标是使这种误差值小于 0.05。

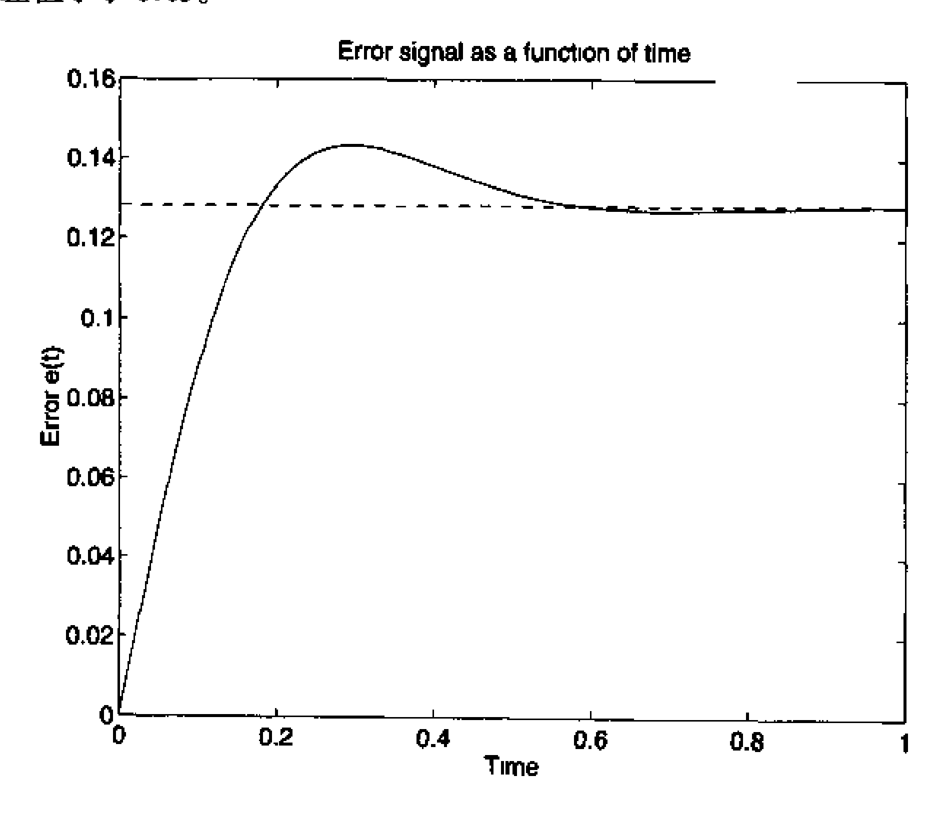


图 10.10 随时间变化的控制系统随动误差

对于状态空间系统，MATLAB 函数：

```
initial(sys,x0)
```

在非零初始条件下进行仿真，其中 *sys* 是被研究的系统，$x0$ 是初始条件向量。假设直流电机在初始位置 $\theta(0)=0$，初始电流 $i(0)=0$，初始角速度 $\omega(t)=5$ rad/s，此外，还假定增益为 2，若线圈上的电压保持在 0 伏（通过短路实现），那么下面的程序将计算电机在初始状态 $x0$ 下的响应曲线。

```
x0 = [0;5;0];
[theta,t,x] = initial(feedback(2 * MotorSS,1),x0);
plot(x(:,1),x(:,2));
grid on;
xlabel('\theta(t)');
ylabel('\omega(t)');
title('Phase plot of the controlled motor');
```

其中 $x(:,1)=\theta(t)$, $x(:,2)=\omega(t)$, $x(:,3)=i(t)$。程序执行后结果如图 10.11 所示。

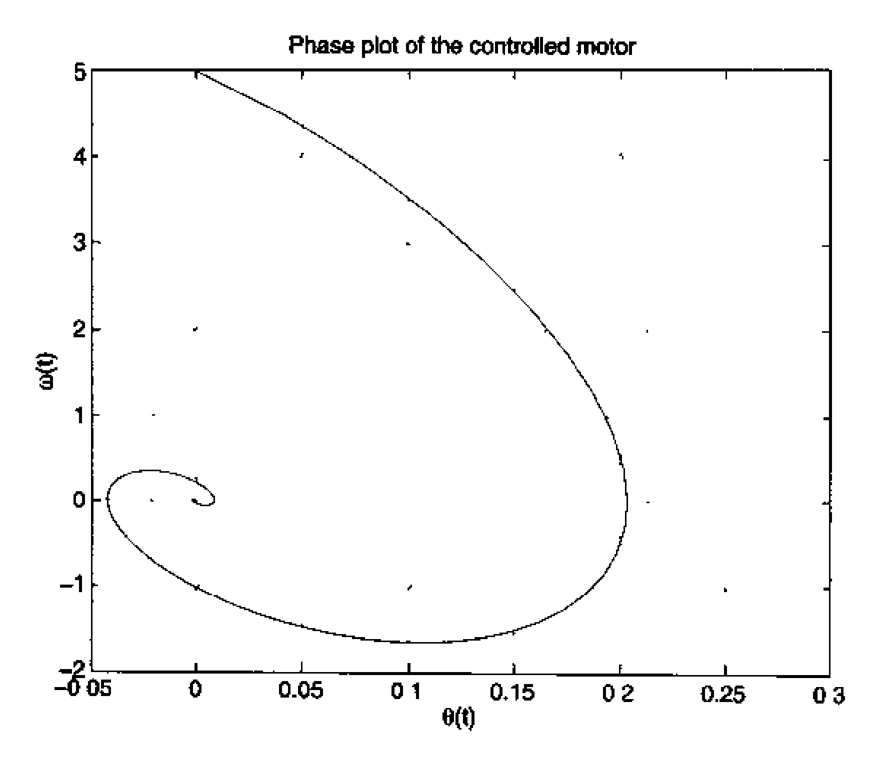


图 10.11 初始角速度 $\omega(t)=5$ rad/s 的转子相位曲线

10.3.2 根据零极点估计系统响应

当用 MATLAB 解决控制问题时，重要的是能定性地判断微分方程的解。由于控制器设计是一个反过程，所以，定性判断能让设计者知道为了达到设计要求应采用什么样的系统。MATLAB 通过求解各种类型的方程可以提高对微分方程求解的定性理解程度。

本节将计算许多系统的阶跃响应，这些系统具有不同的零极点。首先研究具有一个极点没有零点的一阶系统。假设极点位置为 $-\sigma$，则传递函数为：

$$H(s)=\frac{\sigma}{s+\sigma}$$

分子设为 σ 是为了保证系统的直流增益为 1。极点由慢至快从 $\sigma=0.1$ 变化到 $\sigma=2$，下面的程序完成这个计算过程，产生一系列阶跃响应曲线。

```
t = 0:0.1:10;
polevect = 0.1:0.1:2
hold on;
for i = 1:length(polevect)
 y = step(tf([polevect(i)],[1 polevect(i)]),t);
 polt(t,y)
```

```
end;
xlabel('Time');
ylabel('Step response');
title('Step response for a variety of pole locations');
text(0.2,0.95,'p=2');
text(3,0.1,'p=0.1 ( \ Deltap=0.1)')
```

程序执行后的结果如图 10.12 所示。

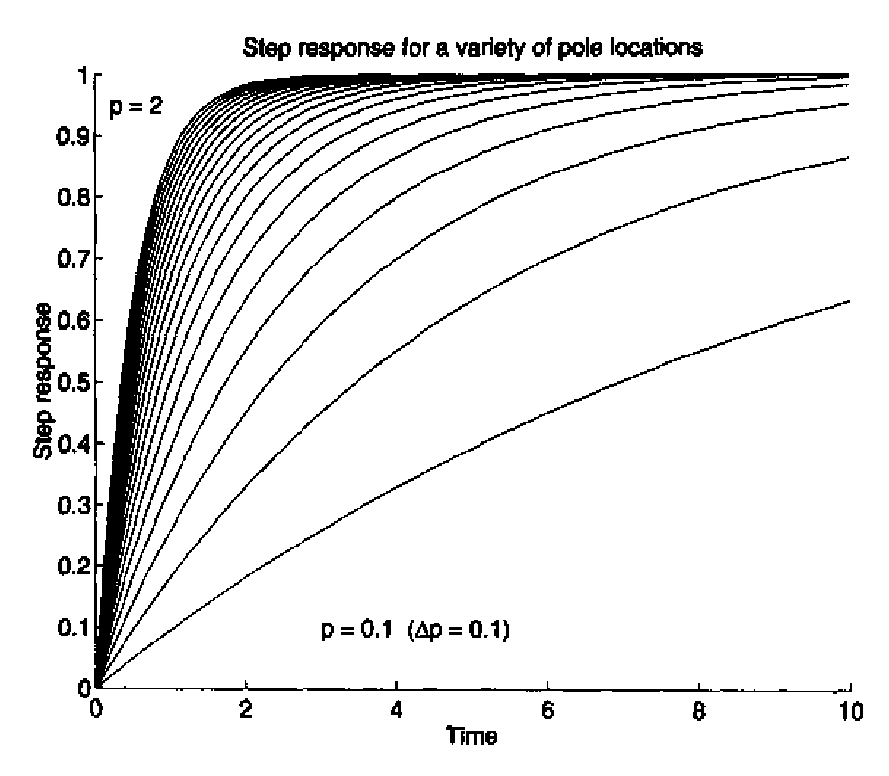


图 10.12 极点 p 变化情况下的一阶系统的响应曲线

当一阶系统的极点接近虚轴(即 σ 变得很小)时,控制系统会变得非常迟钝。通常这种迟钝对控制系统来说并不好,但有时缓慢响应能抑制一些扰动,这种迟钝反而变成好事。因此,系统极点的最佳位置取决于控制系统的设计目标。

若把两个一阶系统串联起来,则形成了一个二阶系统。许多机械系统都是二阶的,因此掌握好二阶系统是非常重要的。研究下面的二阶系统:

$$H(s)=\frac{\sigma^2+\omega^2}{s^2+2s\sigma+\sigma^2+\omega^2} \tag{10.7}$$

系统极点为 $-\sigma\pm j\omega$,其中 σ 是极点的实部,而 ω 是极点的虚部。通常,σ 表示系统的衰减,ω 为机械储能或弹性。因为在此关心的是系统响应与极点的依赖关系,所以传递函数被极点的位置参数化。下面的程序研究极点位置变化对系统响应的影响。首先,设 $\omega=1.0$,以 σ 为变量,画出阶跃响应;然后再设 $\sigma=0.5$,以 ω 为变量,画出阶跃响应。而且在每一个响应旁边画出了极点的轨迹。

```
t=0:0.4:10;sigma=linspace(0.05,1.0,10);
data=zeros(length(t),length(sigma));
omega=1.0;
for i=1:length(sigma)
 data(:,i)=step(tf[sigma(i)^2+omega^2],...
```

```
                          [1 2 * sigma(i)sigma(i)^2 + omega^2]),t);
end;
subplot(2,2,1);
mesh(t, - sigma,data');
ylabel(' \ sigma');
xlabel('Time');
zlabel('Response');
title('Response as a function of \sigma: \omega = 1 0');
data = zeros(length(t),length(omega))
sigma = 0.5;omega = linspace(0.3,2.0,10);
for i = 1:length(omega)
 data(:,i) = step(tf[sigma^2 + omega(i)^2],[1 2 *  sigma sigma^2 + omega(i)^2]),t);
end;
subplot(2,2,3);
mesh(t,omega,data');
ylabel(' \ omega');
xlabel('Time');
zlabel('Response');
title('Response as a function of \ omega: \ sigma = 0.5');
subplot(2,2,2);
hold on;
plot([ -0.1, -0.1],[1.0, -1],'x');
plot([ -0.1, -1],[1.0,1.0]);
plot([ -0 1, -1],[ -1 0, -1.0]);
plot([ -1, -1],[1.0, -1],' < ');
plot([ -2 1],[0 0],'k')
plot([0 0],[ -2 2],'k');
axis([ -2 1  -2 2]);
xlabel('Real axis');
ylabel('Imaginary axis');
title('Pole location');
subplot(2,2,4);
hold on;
plot([ -0.5, -0.5],[0.3, -0.3],'x');
plot([ -0.5, -0.5],[0.3,1.5]);
plot([ -0.5, -0.5],[ -0.3, -1.5]);
plot([ -0.5],[1.5],'^');
plot([ -0.5],[ -1.5],'v');
plot([ -2 1],[0 0],'k');
plot([0 0],[ -2 2],'k');
axis([ -2 1  -2 2]);
xlabel('Real axis');
ylabel('Imaginary axis');
title('Pole location');
```

程序执行后,结果如图 10.13 所示。

像一阶系统那样,稳态响应为 $H(0)=1$。如图 10.13 所示,当 σ 固定,而 ω 增长时,系统响应衰减很小。当 ω 固定,而 σ 增长时,系统响应衰减很大。如果极点的实部接近虚轴,那么系统衰减也很小。如果极点进入复平面的右半平面,则系统响应是无限大的。

电机系统是高阶系统,一般具有二阶系统特性,有时也具有一阶系统特性。在系统极点中,如果有一些极点的实部比其附近的极点实部大很多,且附近没有传输零点,则这些极点将决定系统的输入输出特性。这样就允许设计者用一阶或二阶系统近似表示系统的闭环特性。

为理解这些，要研究直流永磁电机的速度控制问题。运动方程与式(10.2)相同，只是输出用转子的角速度 $d\theta(t)/dt=\omega(t)$ 代替了角度 $\theta(t)$。程序如下：

```
PlantRPM = MotorSS;
PlantRPM.c = [0 1 0];
PlantRPM = minreal(PlantRPM);
set(PlantRPM,'OutputName',' \ omega');
```

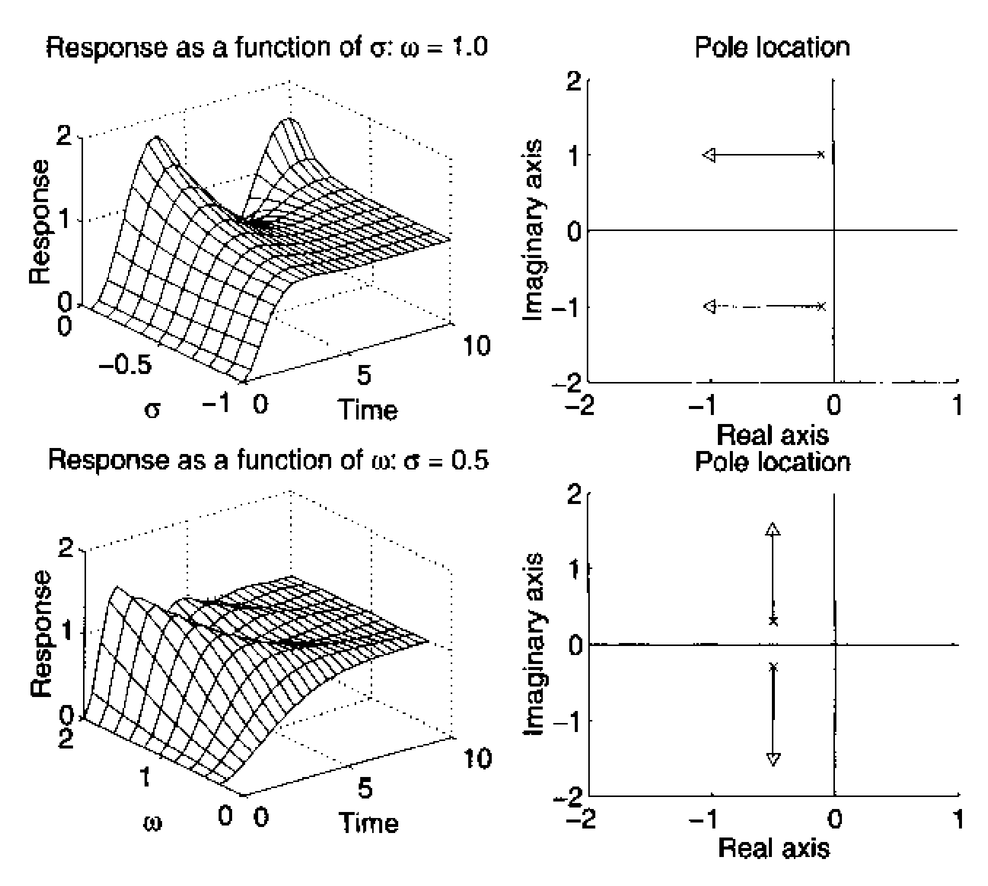


图 10.13　极点位置变化的二阶系统响应曲线

程序的第一行创建系统 PlantRPM，然后读出矩阵 C 确定 $\omega(t)$ 而不是 $\theta(t)$。由于转子角度 $\theta(t)$ 已经是不可观测的了，所以调用 minreal 从方程中消去了这个状态量。键入：

```
pole(plantRPM)/(2 * pi)
```

在 MATLAB 命令窗口中返回系统的极点：一个快速极点约在 160 Hz 位置，它与线圈有关；另一个约在 2 Hz 位置，是慢速极点，与转子运动有关。因为快速极点比慢极点的速度大 10 倍还多，所以由转子运动而引起的慢速极点决定系统的开环响应。设转子速度 $\omega(t)$ 和线圈电流 i 都在 -1 和 +1 之间变化，在此条件下程序进行仿真，并将结果绘在同一个相位曲线图中，即 $\omega(t)$ 与 $i(t)$ 的曲线图。时间没有明显标在相位图上，所以当全部轨线时间超过 5% 时，在轨线上标出一个小圆圈。

```
PlantRPM = MotorSS;
PlantRPM.c = [0 1 0];
PlantRPM = minreal(PlantRPM);
set(PlantRPM,'OutputName',' \ omega');
t = linspace(0,0.05,500);
```

```
hold on;
for i=0:13;
 x0=[-1.4+0.2*i,1];
 [y,t,x]=initial(PlantRPM,x0,t);
 plot(x(:,2),x(:,1));
 k=floor(0.05*length(t));
 plot(x(k,2),x(k,1),'ro');
end;
for i=0:13;
 x0=[-1+0.2*i,-1];
 [y,t,x]=initial(PlantRPM,x0,t);
 plot(x(:,2),x(:,1));
 k=floor(0.05*length(t));
 plot(x(k,2),x(k,1),'ro');
end;
plot([1 -1],[0 0],'k--');
plot([0 0],[1 -1],'k--');
axis([-1 1 -1 1]);
ylabel('i(t)');
xlabel('\omega(t)');
title('Phase portrait of DC motor');
```

系统的相位图如图10.14所示。注意每条曲线的大部分时间处于一维子空间。这个子空间与慢速极点有关系。通常,这些实部很大的极点决定系统的阶跃响应。

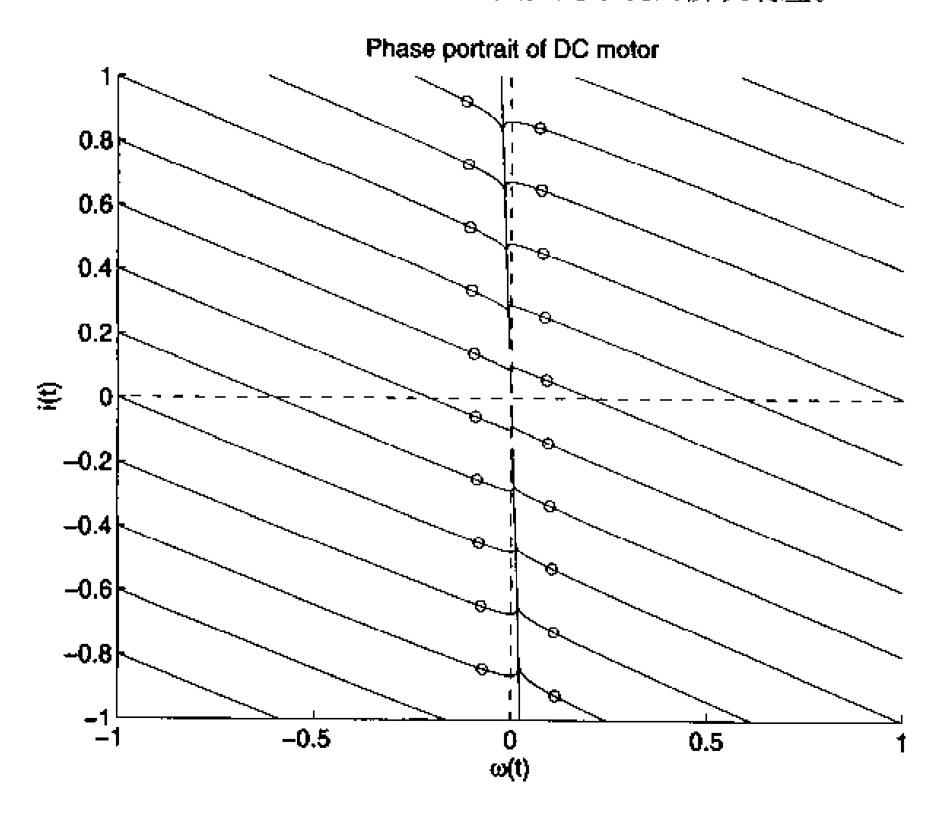


图10.14 直流电机的相位图

尽管零点位置不能通过反馈来改变,但是如果这些零点在极点附近或虚轴附近,那么它们的位置影响还是很重要的。研究二阶系统:

$$H(s)=\frac{-(s-z)}{z(s^2+0.5s+1)}$$

其中 z 是零点位置。一组控制对象的阶跃响应曲线如图 10.15 所示，这些对象的零点有的靠近虚轴，有的穿越虚轴，仿真程序如下。当 $z=-5$ 和 $z=-1$ 时，零点是稳定的；然而在 $z=1$，$z=5$ 的情况下，零点是不稳定的。

```
t=linspace (0,25,200);
Z=[-5 -1 1 5];
Den=[1 0.5 1];
y=zeros(length(t),length(Z));
for i=1:length(Z)
 y(:,i)=step(-1/Z(i)*tf([1 -Z(i)],Den),t);
end
plot(t,y(:,1));
hold on;
plot(t,y(:,2),'K--');
plot(t,y(:,3),'r-o');
plot(t,y(:,4),'g-+');
legend('zero at -5','zero at -1','zero at 1'zero at 5');
xlabel('Time');
ylabel('Step response');
title('The effect of the position of the zero');
```

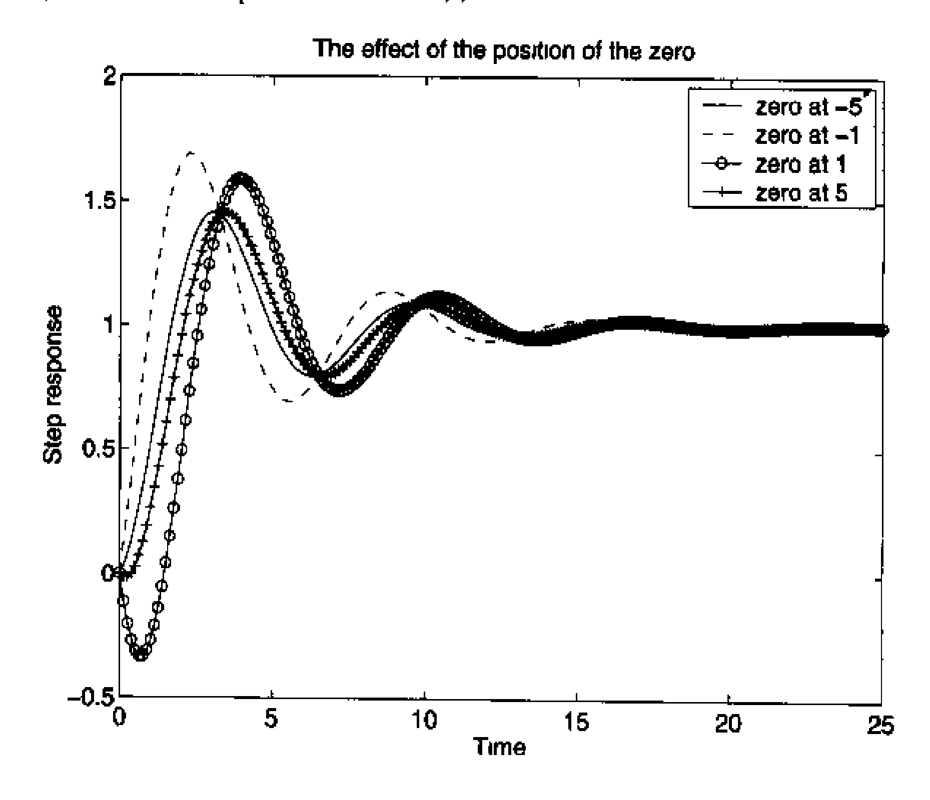


图 10.15 临近和穿越虚轴零点产生的影响

尽管不稳定零点不会使系统不稳定，但是它们限制了反馈量。如图中位置为 1 的零点是不稳定零点，它会使系统性能恶化。具有一个或多个不稳定零点的系统称为非最小相位系统。

从观测的角度或从自激的角度来看，零点可以消除形式上的不稳定。研究下式给出的四种不同阶次的系统。

$$G_1=\frac{1}{s+1} \qquad G_3=G_1(s)*G_2(s)$$

$$G_2 = \frac{100}{s^2 + 10s + 100} \quad G_4 = \frac{19.8s + 20}{s + 20} G_3(s)$$

对象 G_3 的阶跃响应与前述的一阶对象 G_1 的响应非常类似，因为极点 -1 为决定性复极点。但是，若在极点 -1 附近有一个零点，那么极点将不起决定性作用。由 G_4 确定的系统在 G_3 极点 -1 附近增加的零点掩盖了它的作用。然而这在实际中一般不能做到，因为要使 G_3 快速响应，则 G_4 首先应产生很大输出，这可能使执行器饱和或损坏。从 G_4 的阶跃响应中可以看到这一点。通过下面的程序可得到 G_3 和 G_4 的阶跃响应曲线，如图 10.16 所示：

```
G1 = tf([1],[1 1]);
G2 = tf([100],[1 10 100]);
G3 = G1 * G2;
G4 = tf([19.8 20],[1 20]) * G3;
t = linspace(0,6,200);
yG3 = step(G3,t);
yG4 = step(G4,t);
plot(t,yG3,'k--',t,yG4,'k-')
legend('Response of G3','Response of G4')
xlabel('Time')
ylabel('Response')
title('Zero masking a slow pole')
```

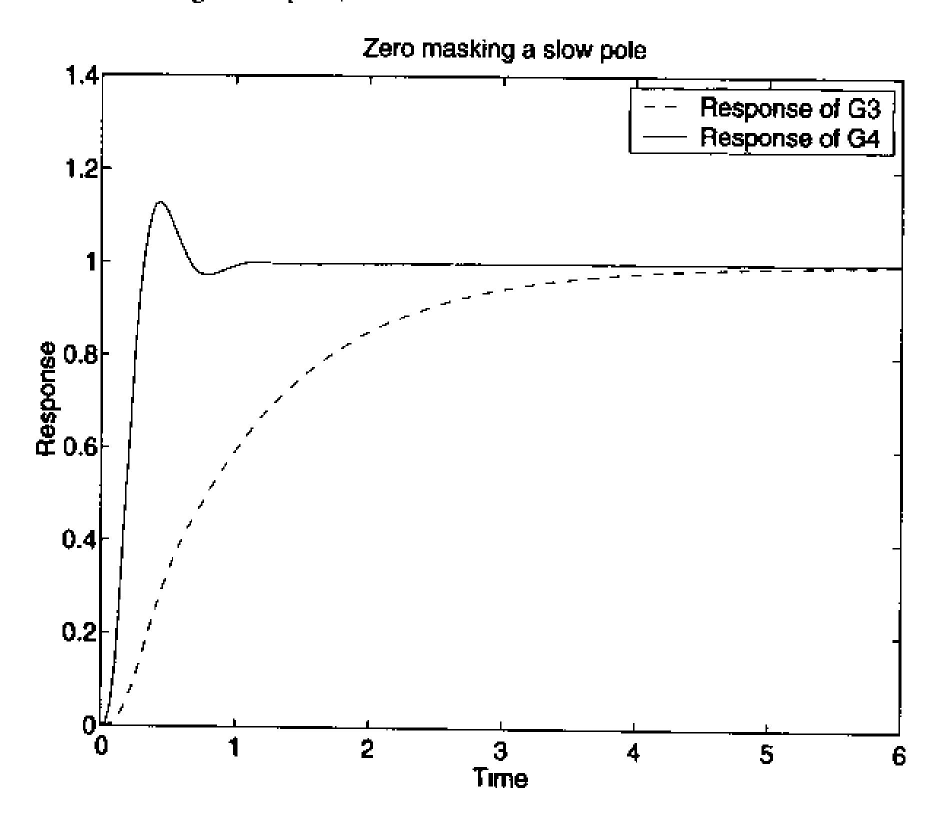


图 10.16 零点隐藏了系统的缓慢动态特性

利用零点和极点之间的距离可以度量输入或输出矩阵与特征向量的正交性。如果零点在极点的正上方，那么它的模在输出端是不可激励的或不可见的。所谓能控性和能观性就是用于描述这种现象的。

10.4 设计工具

本节讨论 MATLAB 中的设计工具及设计准则。在计算机普及之前，许多设计都是采用图解的方法，图解方法包括：

bode——波特图法
nyquist——奈奎斯特法
rlocus——根轨迹法

还有基于计算机的现代设计方法，如 lqr 和 lqe。

在设计过程中，有一个准确的对象模型是非常重要的。设计一个控制系统需要在对象模型的建立及简化上付出很多努力。建模过程不是本章讨论的重点。MATLAB 提供了 ident 工具箱，它能够接收实验数据并给出相应的最佳模型。

控制系统设计准则包括三个要求：

- 稳定性
- 暂态特性
- 稳态特性

在解微分方程时，显然需要稳定性。分母多项式的每一个根对应着方程的一个解，并且任何一个具有正实部的根都会使系统响应按指数规律增加。虚轴上的极点使系统处于临界稳定状态。后面两个设计准则必须建立在稳定的条件下才可进行研究。暂态特性要求观察系统单位阶跃响应的短期行为，而稳态特性为长期行为，主要用于研究下列输入时的随动误差，可能的输入是阶跃、斜坡或极少用到的抛物线。稳定性准则经常在系统中遇到，一些系统开始时是不稳定的，但最终一定会稳定。倒置摆和磁性轴承都属于这类系统的例子。许多系统利用反馈使其成为稳定系统，闭环控制系统的稳定性可以很容易地用闭环系统传递函数分母上的根来检测。任何具有正实部的根都表明闭环系统是不稳定的。

10.4.1 设计准则

本节以直流电机为例说明设计工具，并讨论直流电机闭环系统的稳定性。选用比例控制器，它输入期望位置和实际位置的差，并乘以增益。设计者的主要任务是选择比例控制器的增益。下面的程序产生图 10.17，图中表明增益参数从 1 变化到 200 时的右侧极点实部曲线。

```
gains = linspace(1,200,50);y = [];
for i = 1:length(gains)
 y = [y,max(real(pole(feedback(gains(i) * MotorSS,1))))];
end;
plot(gains,y,[0 200],[0 0],'r--');
xlabel('Proportional controller gain');
ylabel('Real part of right-most pole');
title('Loss of stability for a DC motor system');
```

当增益较小时系统是稳定的；但是当增益远大于 128 时，闭环系统响应将是无限的，此时电机可能被损坏。

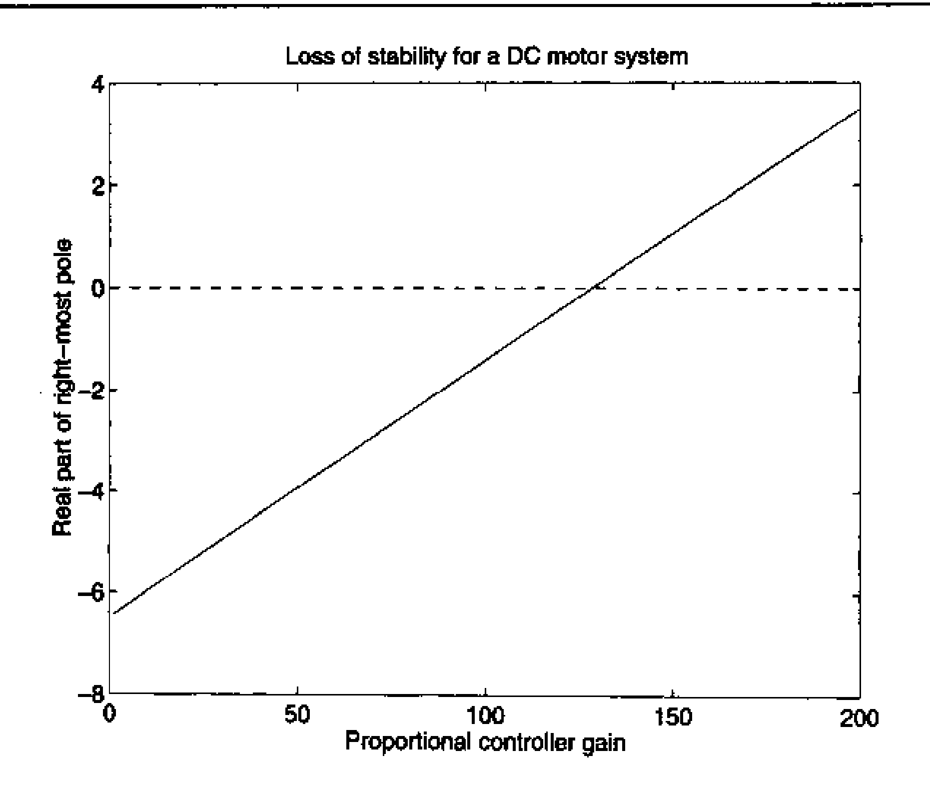


图 10.17 闭环系统右侧极点实部作为控制器增益函数

由下面的函数产生图 10.18 所示的波特图：

```
bode(MotorSS)
```

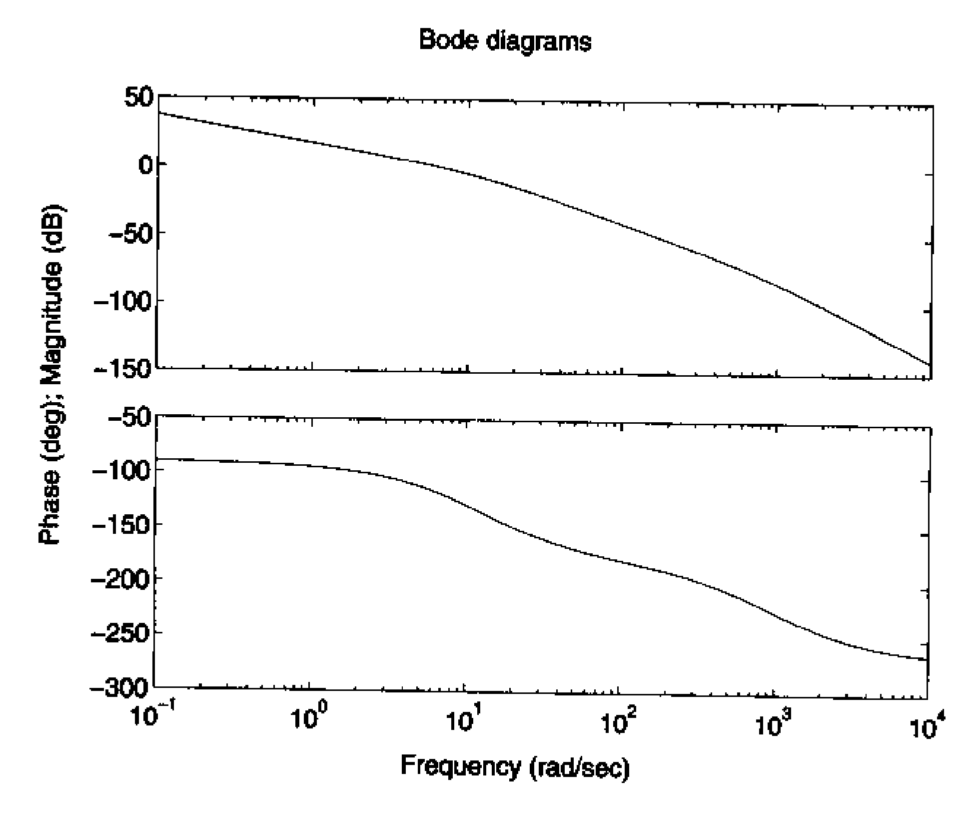


图 10.18 给定相角余量和幅值余量条件下描述截止频率的波特图

并且可确定在多大增益时系统不稳定。命令：

```
[gm,pm,wgm,wpm] = margin(MotorSS);
```

用来计算幅值余量(*gm*)和相角余量(*pm*)以及与之相对应的频率 *wgm* 和 *wpm*。在此,幅值(余量)(在系统变得不稳定之前的幅值)是 *gm* = 128.37 dB 或 42.2 dB。

即使闭环系统是稳定的,但其闭环特性也可能不令人满意。直流永磁电机系统对一系列稳定增益的阶跃响应曲线如图 10.19 所示,可由下面的程序得到,时间范围是 0 到 1 s。四个增益在 1 到 10 之间按对数比例等间距分布。

```
t = linspace(0,1,100);
gains = logspace(0,1,4);
hold on;
for i = 1:length(gains)
 [y,t] = step(feedback(gains(i) * MotorSS,1));
 plot(t,y);
end;
xlabel('Time');
ylabel('Response');
title('Step response with various gains');
text(0.16,1.5,'gain = 10');
text(0.16,0.4,'gain = 1');
```

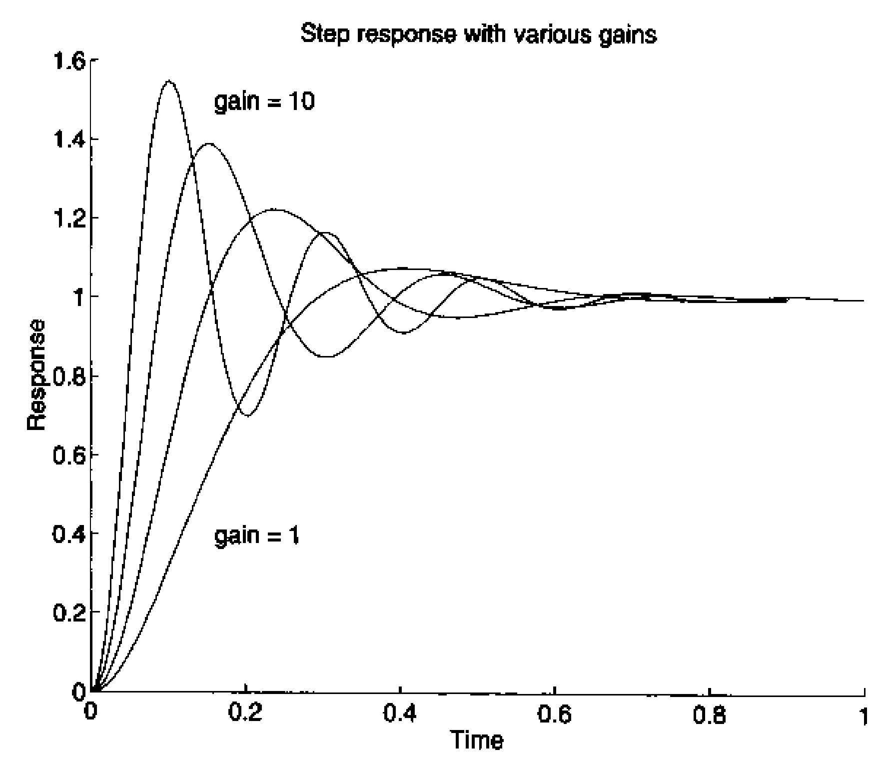


图 10.19 比例控制的电动机系统阶跃响应

随着增益的变大,响应速度加快,高增益的控制器会出现超调,并在预定值附近产生振荡。有很多性能指标可以用来量化这种现象,如:上升时间、超调和调节时间等。

暂态特性可以通过系统的阶跃响应测量。上升时间是由终值的 10% 上升到 90% 时系统所需的时间。与之相关的值是峰值时间,即系统第一次达到峰值的时间。在特定系统中,上升时间与峰值时间在设计上与超调是相互矛盾的。超调量是指系统超过目标值的大小。调节时间一般定义为系统从开始到进入偏离终值 2% 的范围内所用的时间。下面的 *Transient* 函数计

算这些参数。若系统不稳定,则函数返回[-1,-1,-1]。

```
function criteria = transient(system)
criteria = [-1 -1 -1];
maxP = max(real(pole(system)));
if maxP >= 0
 return
end
MaxTime = -6 * (1/maxP);
Time = linspace(0,MaxTime,500);
Response = step(system,Time);
[ResponseMax,IndexMax] = max(Response);
FinalValue = Response(end);
TimeLow = interp1(Response(1:IndexMax),Time(1:IndexMax),0.1 * FinalValue);
TimeHigh = interp1(Response(1:IndexMax),Time(1:IndexMax),0.9 * FinalValue);
criteria(1) = TimeHigh - TimeLow;
k = length(Time);
while(k > 0)&(0.02 > abs((FinalValue - Response(k)/FinalValue));
 k = k - 1;
end;
criteria(2) = Time(k);
criteria(3) = 100 * (max(Response) - FinalValue)/FinalValue;
```

其中 *criteria*(1) = 上升时间, *criteria*(2) = 调节时间, *criteria*(3) = 超调量。

用这个函数估算直流电动机的控制器,调用:

```
transient(feedback(MotorSS,1));
```

显示向量[0.1935, 0.6080,7.6136]。其中 0.1935 是上升时间,0.6080 是调节时间,而 7.6136 是超调量。

10.4.2 设计方法

对于第 10.4.1 节讨论的设计准则,将介绍一些设计方法,并举例说明这些方法在电动机控制中的应用。常用的三种不同方法是:

- 基于频率的方法
- 根轨迹法
- 基于 LQG 的方法

基于频率的设计方法不需要精确的模型,只需要一批实验结果。其他两种方法都需要精确的模型。

现在研究电动机控制器问题。设计准则要求超调量保持在 20% 以下,因此图 10.19 所示的许多增益条件都是不被采用的。此外,还要求闭环系统能快速响应,上升时间小于 0.05 s。若直接用比例控制器,将会出现问题。一方面,为了使上升时间小于 0.05 s,增益必须远大于 3;另一方面,为使超调小于 20%,增益必须远小于 2。下面的程序产生图 10.20,图中显示了对于比例控制器超调量和上升时间随增益变化的曲线。

```
kp = 0.4 * logspace(0,1,20);result = [];
for i = 1:length(kp)
```

```
result = [result;transient(feedback(kp(i) * MotorSS,1))];
end;
[ax,h1,h2] = plotyy(kp,result(:,3),kp,result(:,1));
xlabel('Controller gain');
ylabel('Percent overshoot');
v = axis;
text(v(2) * 1.06,v(3) + (v(4) - v(3))/2.4,'Rise time','rotation',90);
set(h2,'Marker','s');
set(h1,'Marker','o');
title('Percentage overshoot and rise time as a function of gain');
```

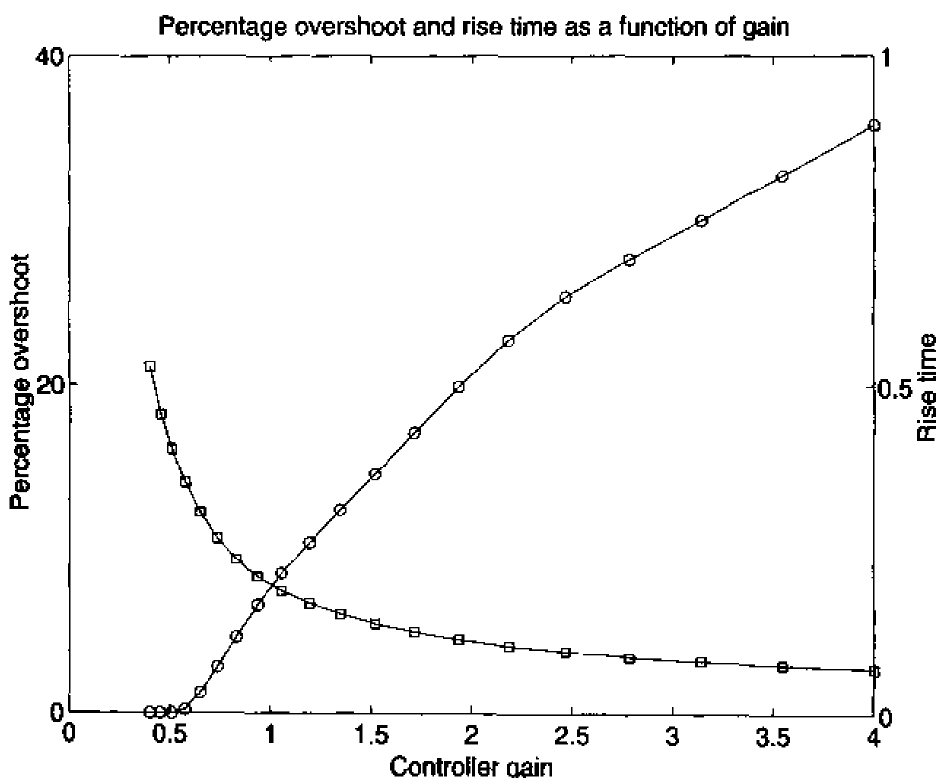


图 10.20 控制器增益函数的超调量(圆圈)和上升时间(方块)曲线

首先采用基于频率的设计方法。超调的质标要求可以转化为相角余量的质标要求。对这个系统而言,相角余量为45°时,即可满足超调量要求。从图10.18的波特图中可以看出,当相角为-135°(相角余量为45°)时,允许的增益近似为2.0。用瞬态下的fzero函数更精确地计算增益。首先创建函数PEcontrol。

```
function s = PEcontrol(gain)
rval = transient(feedback(gain * MotorSS,1));
s = rval(3) - 20;
```

然后使用:

```
options = optimset('display','off');
gain = fzero('PEcontrol',2,options)
```

得到超调量为20%条件下的增益为1.938 4。用命令:

```
transresp = transient(feedback(1 9384 * MotorSS,1))
```

得出 *transresp*(1) = 0.111 s,上升时间比设计指标慢一倍多。

一般采用超前控制器来提高系统的暂态响应,适当的设计超前控制器能在小的频率范围内增加相位;同时,相位增加允许使用更大的增益。超前控制器的零点选在 - 15,刚好在第二个开环极点 - 13 的左边。这样能确保超前控制器相位增加,与直流电动机相位下降处于同一个位置。超前控制器的极点在 - 100,几乎是零点位置的 10 倍。理论上极点位置越靠左越好,但是极点太靠左会使控制器对噪声很敏感。经验法则是极点位置不能超过零点位置的 10 倍。最后的控制器传递函数是:

$$C(s)=\frac{100}{15}\frac{s+15}{s+100}=\frac{6.667s+100}{s+100}$$

将传递函数乘以 100/15,使直流增益为 1。

下面的程序产生图 10.21,用来比较未补偿系统和补偿系统的频率响应:

```
Control = tf([6.667 100],[1 100]);
bode(MotorSS,'k-',Control * MotorSS,'k--')
```

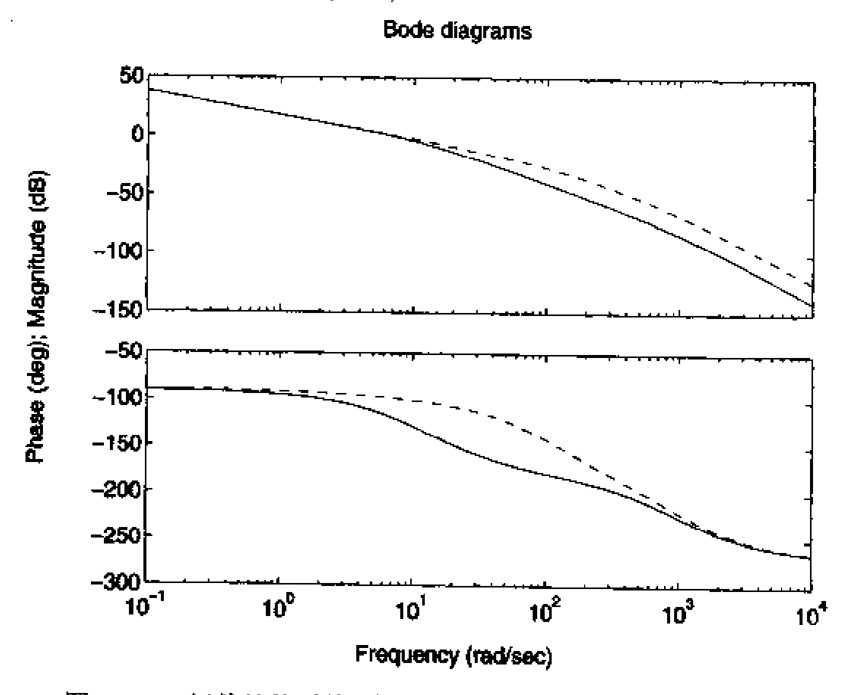


图 10.21 超前补偿系统(虚线)和未补偿系统(实线)的波特图

注意高频时相位下降是如何被滞后的。使用波特图找到恰当的反馈增益近似为 10。为了计算超调量为 20% 时的精确增益,创建函数 LDcontrol。

```
function s = LDcontrol(gain)
rval = transient(feedback(gain * tf([6.667 100],[1 100]) * MotorSS,1));
s = rval(3) - 20;
```

然后用:

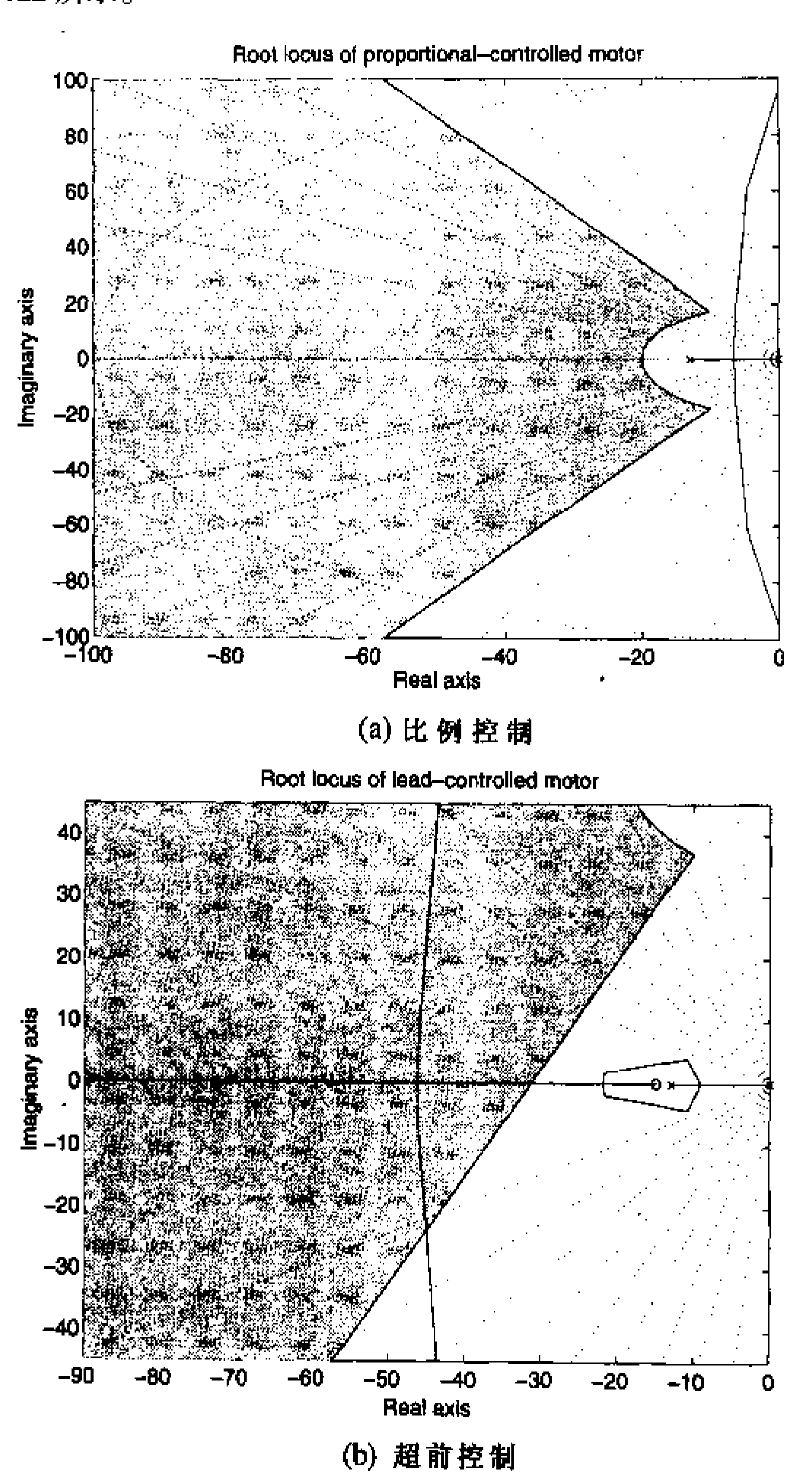
```
options = optimset('display','off');
gain = fzero('LDcontrol',[10],options)
```

得到超调量为 20% 时的 $gain$ = 13.010 8。用:

```
tansresp = transient(feedback(13.0108 * tf([6.667 100],[1 100]) * MotorSS,1));
```

可以得到 $transresp(1)$ = 0.017 0 s,上升时间比期望值几乎快 2 倍。比例和超前控制的系统阶跃

响应曲线如图 10.22 所示。



(a) 比例控制

(b) 超前控制

图 10.22　电动机位置系统根轨迹图

用超前补偿器可以同时满足超调和上升时间的要求。基于频率的设计不依靠精确的模型而是从波特图中得到数据。其他基于频率的设计方法还包括奈奎斯特曲线和尼科尔斯(*Nichols*)图,使用方法与前述方法类似。

根轨迹是另一种常用的设计方法,同样是用超前补偿但方法不同。给定复数极点所对应的角度 ξ 及半径 ω_n,为:[①]

① 参见 D. K. Anand 和 R B. Zmood, 出处同上, 1995 年。

$$M_p = \exp\left(-\frac{\pi\zeta}{\sqrt{1-\zeta^2}} \right)$$

$$T_r \approx \frac{1}{\omega_n}(1 + 1.4\zeta)$$

其中 M_p 为峰值，T_r 为上升时间。转换这些约束条件以满足动态设计要求，并将闭环极点配置在复平面中。超调量小于 20% 的设计要求限制了 ζ，要求主导极点位于以负虚轴为中心的 120°楔形内。上升时间 0.05 s 大约对应最小的极点半径 $\omega_n = 20$。这些都是经验法则，它们起着很重要的作用。下面的程序首先确定闭环极点在复平面上的区域，然后画根轨迹检验是否满足条件。

```
theta = linspace( -2/3 * pi, -4/3 * pi,15);
X = [20 * cos(theta)200 * cos( -4/3 * pi)200 * cos( -2/3 * pi)20 * cos( -2/3 * pi)];
Y = [20 * sin(theta)200 * sin( -4/3 * pi)200 * sin( -2/3 * pi)20 * sin( -2/3 * pi)];
hold on;
fill(X,Y,'y');
sgrid;
rlocus(MotorSS);
axis(100 * [ -1 0  -1 1]);
ylabel('Imaginary axis');
xlabel('Real axis');
title('Root locus of proportional-controlled motor');
```

结果如图 10.22(a)所示。类似的超前控制器系统可以用下面的程序产生：

```
theta = linspace( -2/3 * pi, -4/3 * pi,15);
X = [20 * cos(theta)200 * cos( -4/3 * pi)200 * cos( -2/3 * pi)20 * cos( -2/3 * pi)];
Y = [20 * sin(theta)200 * sin( -4/3 * pi)200 * sin( -2/3 * pi)20 * sin( -2/3 * pi)];
hold on;
fill(X,Y,'y');
sgrid;
rlocus(MotorSS * tf([6.667,100],[1 100]));
axis(90 * [ -1 0  -0.5 0.5]);
ylabel('Imaginary axis');
xlabel('Real axis');
title('Root locus of lead-controlled motor');
```

由此可见，控制器和控制对象 *MotorSS* 交换了它们的计算次序，理论上，这种计算次序的变化不会影响根轨迹。然而，当超前控制器的根轨迹命令先于对象命令执行时，计算过程将十分缓慢，并且可能会导致某些系统死锁。这是由于控制器零点与对象极点靠得太近的缘故。为简化计算闭环极点的过程，需要对控制器的形式进行转化。如果先计算控制器后计算对象，则这种转化将是奇异的，也就是级联是不可控的。如果变化一下计算次序，则系统变为不可观测的，也就不影响控制器形式的转化。

程序的执行结果如图 10.22(b)所示。应注意到对于比例控制器而言，在希望的闭环根轨迹区域内没有增益。超前控制器的基本作用是将根轨迹线拉回到满意的极点区域内。为了找到适当的增益，使用：

```
rlocus(MotorSS * tf([6.667 100],[1 100]))
[k,p] = rlocfind(MotorSS * tf([6.667 100],[1 100]))
```

将十字线放在根轨迹与可接受边缘的交叉点上。函数 rlocus 放在 rlocfind 之前是因为 rlocfind 不画根轨迹，这个过程产生增益 12。这些区域是近似的，因此它是一种通过仿真微调增益的实用方法。从仿真中可以找到可用的最高增益为 13。

在根轨迹设计方法中，目的是将闭环极点放在满意的区域内。针对于状态空间模型开发的线性代数方法可以将极点直接放在期望的位置就可解决这类问题。这里介绍的两种极点配置方法采用：

```
place
```

和

```
acker
```

而 LQG 设计方法采用：

```
lqr
lqe
```

和

```
reg
```

函数。

为了满足设计指标选择极点位置为 -30、$-20+30i$ 和 $-20-30i$。这些极点位置是任意选择的，但最好在图 10.22 中的阴影区域内。下面的程序用 place 计算增益矩阵 $\boldsymbol{K}$，以使得矩阵 $(\boldsymbol{A}-\boldsymbol{BK})$ 的特征值在期望的位置。矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 如式(10.3)所示。

```
DesiredPoles = [ - 30, - 20 + 30 * i, - 20 - 30 * i];
[A,B,C,D] = ssdata(MotorSS);
K = place(A,B,DesiredPoles)
```

程序执行后给出 $\boldsymbol{K}=[0.3900\ \ -0.1002\ \ -4.6517]$。假设给定系统的内部状态变量为 x，所需要的反馈为 $u=Kx$。因为上式只能得到输出，所以还需要设计状态观测器。计算控制器和观测器的完整程序清单如下：

```
DesiredPoles = [ - 30, - 20 + 30 * i, - 20 - 30 * i];
[A,B,C,D] = ssdata(MotorSS);
K = place(A,B,DesiredPoles);
L = (place(A',C',3 * DesiredPoles))';
ControlSS = reg(MotorSS,K,L);
clSys = feedback(MotorSS,ControlSS, + 1);
clSys = 1/dcgain(clSys) * clSys;
step(clSys);
```

作为状态空间控制器，程序的执行结果如图 10.24 所示。程序的前 3 行产生反馈增益矩阵 $\boldsymbol{K}$。在能观性和能控性之间进行对偶变换，用 place 函数计算反馈观测增益矩阵 $\boldsymbol{L}$。这个矩阵依赖于系统矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{C}$，以及一些期望的观测器极点位置，在此设为 3 倍于反馈极点的位置。函数 reg 使用对象输出创建了一个估算器，用于估算内部状态并输出正确的值。为了让信号值具有合适的符号应采用正反馈。

前面的例子中，极点位置的选择是满足暂态特性要求的，实际上这些选择并不是惟一的。

因为在一个区域内,合适的极点位置很多,设计时只是在合适的区域内任意选择一些极点位置而已。线性二次高斯控制器的设计方法遵循类似的步骤,但是为设计者分配极点位置提供了系统的方法。极点的选择可以使下面的积分最小化①:

$$J = \int [x'(t)Qx(t) + u'(t)Ru(t)]dt \tag{10.8}$$

其中,$x(t)$为 t 时刻系统的内部状态,$u(t)$为 t 时刻的输入向量,$\boldsymbol{Q}$ 为半正定矩阵,而 $\boldsymbol{R}$ 为正定矩阵。矩阵 $\boldsymbol{Q}$ 经常选为$\boldsymbol{C'C}$,以便使上式的第1项简化为输出误差的方矩阵。

设计者可以通过修改相应的 $\boldsymbol{Q}$ 和$\boldsymbol{R}$ 值来调整相应的状态误差输入。特殊情况下,对象只有一个输入,因此 $\boldsymbol{R}$ 为正的标量。当 $\boldsymbol{Q} = \boldsymbol{C'C}$ 时,下面的程序画出最优极点的位置(作为 $\boldsymbol{R}$ 的函数)。此外,极点所处的区域必须要满足暂态设计要求。

```
[A,B,C,D] = ssdata(MotorSS);
clPoles = [];
R = logspace(-4,1,60);
for i = 1:length(R)
 [K,S,E] = lqr(A,B,C' * C,R(i));
 clPoles = [clPoles,E];
end;
theta = linspace(-2/3 * pi, -4/3 * pi,15);
X = [20 * cos(theta)200 * cos(-4/3 * pi)200 * cos(-2/3 * pi)20 * cos(-2/3 * pi)];
Y = [20 * sin(theta)200 * sin(-4/3 * pi)200 * sin(-2/3 * pi)20 * sin(-2/3 * pi)];
fill(X,Y,'c');
hold on;
plot(real(clPoles),imag(clPoles),'x');
axis(40 * [-1 0 -0.5 0.5]);
sgrid;
title('Optimal root locus for motor system');
ylabel('Imaginary axis');
xlabel('Real axis');
```

程序运行结果的根轨迹如图10.23所示,图中用"x"分散地画出了根的位置,而不采用实线。当 $\boldsymbol{R}$ 值很大时,输入所起的作用相对于输出误差所起的作用要大,因而几乎不需要控制即可满足要求。闭环极点与开环极点临近。如果输入所起的作用很小,则最优闭环极点向复平面右半平面移动,需要引入更多的控制作用,且系统响应会更快。因为要同时满足优化条件和暂态特性要求这两个方面,所以必须选择 $\boldsymbol{R}$ 使最优极点位于阴影区域内。

至此所讨论的三种主要控制器设计方法的阶跃响应曲线由下面的程序产生,并在图10.24中进行比较。比例控制器未能满足上升时间设计要求,而采用极点配置的超前控制器和状态空间控制器都满足了设计要求。状态空间控制器的另一个好处是几乎无超调。

```
t = linspace(0,1,200);
yp = step(feedback(1.9416 * MotorSS,1),t);
yl = step(feedback(13.0108 * tf([6 667 100],[1 100]) * MotorSS,1),t);
DesiredPoles = [-30, -20 + 30 * i, -20 - 30 * i];
[A,B,C,D] = ssdata(MotorSS);
K = place(A,B,DesiredPoles);
L = (place(A',C',3 * DesiredPoles))';
```

① 参见T Kailath,出处同上,1980年。

```
ControlSS = reg(MotorSS,K,L);
clSys = feedback(MotorSS,ControlSS, + 1);
clSys = clSys/dcgain(clSys);
ys = step(clSys,t);
plot(t,yp,'k-',t,yl,'k--',t,ys,'k-.');
xlabel('Time');
ylabel('Step response');
title('Step response of proportional-,lead-,and state-space controllers');
legend('Proportional','Lead','State-space',4);
```

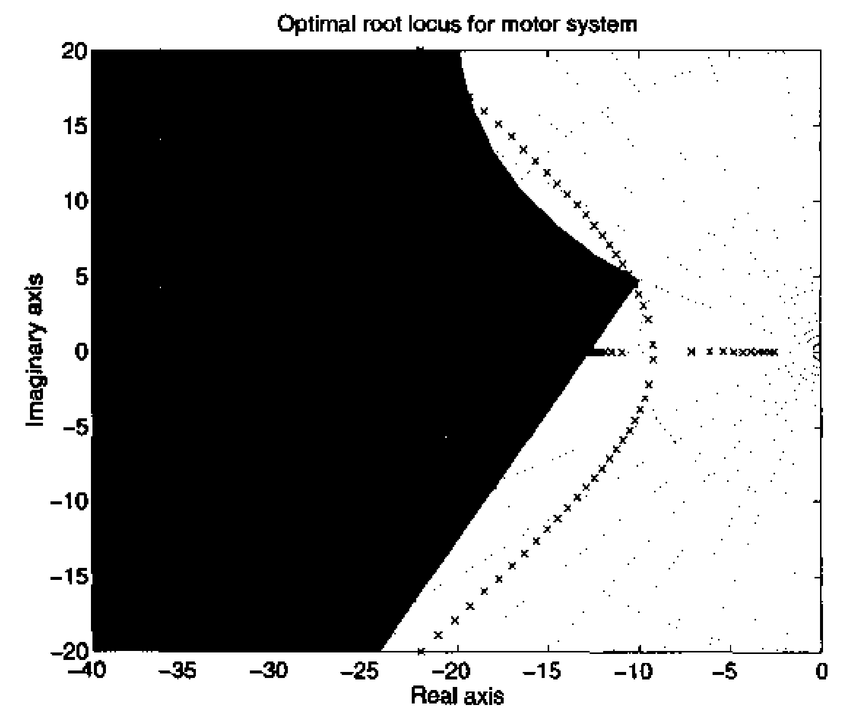


图 10.23 输入代价加权值 R 作为函数时最优控制器的闭环极点

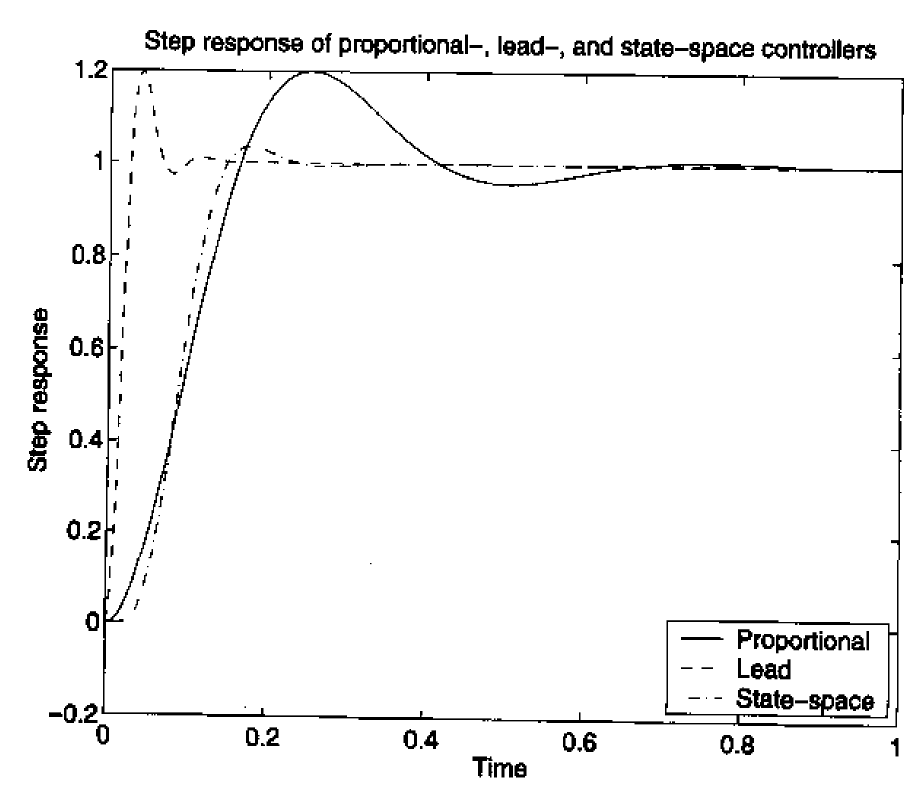


图 10.24 阶跃输入的比例、超前和状态空间控制器的比较

10.5 设计实例

本节将用 MATLAB 工具对4个不同的实际系统设计控制器。

- 带弹性轴的直流电机——设计超前 - 滞后控制器以避免弹性轴产生振动。
- 单轴磁悬浮系统——设计 PID 控制器以保持总体的平衡位置。
- 倒置摆——设计多输入单输出控制器以保持摆的垂直。
- 磁悬浮飞轮——设计多输入多输出控制器以保持飞轮悬浮。

控制器的设计过程包括4个步骤:

1. 确定控制器设计要求
2. 建立对象模型
3. 设计满足要求的控制器
4. 仿真并检验设计的控制器

下面几节中的对象模型由基本原理引伸而来,没有经过实践检验。在控制器的设计中,设计者必须要牢记不能过分强调模型的有效性和精确性。上面所描述的后3种对象模型是非线性的,但并不严重。在工作点附近每个模型都可被线性化,而且是稳定的。由于有些系统开环不稳定,因此从波特图获得开环数据采用基于频率的设计方法仅适用于带有弹性轴的直流电机系统。

下面几节介绍的控制器包括:

- 超前(滞后)
- 超前 - 滞后
- PID
- LQG

在上面所述的每一种方法中,根轨迹都是基本的设计方法。实际上,PID 控制器是目前单入单出系统中最常见的控制器类型。

10.5.1 弹性指针的超前 - 滞后控制

研究硬盘驱动器的读写头。设计目的是让读写头尽可能快地移到期望位置,并为读写操作提供一个稳定的平台。以线圈作为激励,当驱动能力给定时增大移动速度的方法是将固定读写头的旋杆重量减小。旋杆重量的减小使移动加快,但同时也可能使旋杆进入振动状态。

以图10.25所示的带有弹性轴的直流电机作为设计模型。用户指定期望的角度 θ_d 和用来测量旋杆偏角 θ 的控制系统,并要求以尽可能快的方式跟踪命令。然而,若弹性指针到达指定位置后长时间振动,则读写头将不能正常工作。

弹性指针的运动方程为:[①]

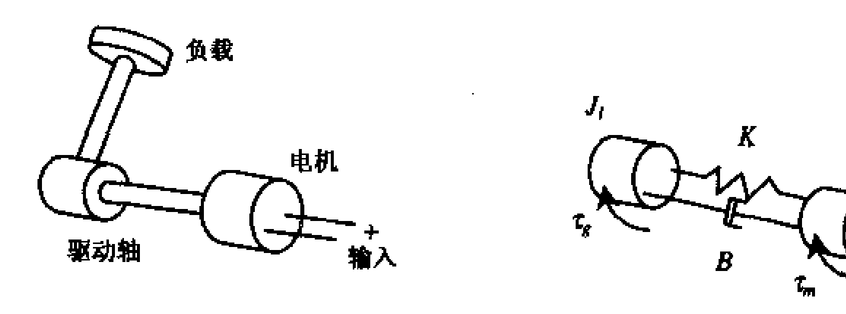
① 参见 R. C. Dorf 和 R. H. Bishop, *Modern Control Systems*, Addison Wesley, Reading, MA, 1998 年。

$$L\frac{\mathrm{d}i(t)}{\mathrm{d}t}+k_b\frac{\mathrm{d}\theta(t)}{\mathrm{d}t}+Ri=v(t)$$

$$J_m\frac{\mathrm{d}^2\theta(t)}{\mathrm{d}t^2}-k_\tau i(t)=b\left(\frac{\mathrm{d}\phi(t)}{\mathrm{d}t}-\frac{\mathrm{d}\theta(t)}{\mathrm{d}t}\right)+k[\phi(t)-\theta(t)] \tag{10.9}$$

$$J_l\frac{\mathrm{d}^2\phi(t)}{\mathrm{d}t^2}=-b\left(\frac{\mathrm{d}\phi(t)}{\mathrm{d}t}-\frac{\mathrm{d}\theta(t)}{\mathrm{d}t}\right)-k[\phi(t)-\theta(t)]$$

其中 θ 为转子的角度，ϕ 为指针的定位，$J_m=0.03\ \mathrm{kg.m^2}$ 为转动惯量，$J_l=0.015\ \mathrm{kg.m^2}$ 为负荷惯量，$b=0.01$ Nm/rad/s 为弹性轴的阻尼系数，$k=10$ Nm/rad 为弹性轴的恢复常数。电感 L、阻尼 R、反向电动势常数 k_b 及电动机力矩常数 k_τ 等各种参数的定义见第 10.2.1 节。



(a) 带弹性驱动轴的指针　　(b) 等价模型

图 10.25　设计模型

通过引入状态变量，可以将耦合的二阶微分方程组转换为线性耦合的一阶方程组：

$$x_1=\theta \qquad x_4=\phi$$

$$x_2=\frac{\mathrm{d}\theta}{\mathrm{d}t} \qquad x_5=\frac{\mathrm{d}\phi}{\mathrm{d}t}$$

$$x_3=i$$

则式(10.9)可转换为：

$$\frac{\mathrm{d}x_1}{\mathrm{d}t}=x_2$$

$$\frac{\mathrm{d}x_2}{\mathrm{d}t}=-\frac{k}{J_m}x_1-\frac{b}{J_m}x_2+\frac{k_\tau}{J_m}x_3+\frac{k}{J_m}x_4+\frac{b}{J_m}x_5$$

$$\frac{\mathrm{d}x_3}{\mathrm{d}t}=-\frac{k_b}{L}x_2-\frac{R}{L}x_3+v$$

$$\frac{\mathrm{d}x_4}{\mathrm{d}t}=x_5$$

$$\frac{\mathrm{d}x_5}{\mathrm{d}t}=\frac{k}{J_l}x_1+\frac{b}{J_l}x_2-\frac{k}{J_l}x_4-\frac{b}{J_l}x_5$$

下面的函数 Pointer 用于计算一阶方程系统矩阵，并返回一个状态空间的系统对象模型。

```
function Plant = Pointer
L = 5e - 3; R = 5; kb = 0.125;
ki = 15; Jm = 3e - 2;
Jl = 0.5 * Jm; k = 10; b = 0.01;
A = [0 1 0 0 0;
     - k/Jm  - b/Jm ki/Jm k/Jm b/Jm;
```

```
    0 -kb/L -R/L 0 0;
    0 0 0 0 1;
    K/Jl b/Jl 0 -k/Jl -b/Jl];
B=[0;0;1/L;0;0];
C=[0 0 0 1 0];
D=0;
Plant=ss(A,B,C,D);
```

系统的输出为指针的角度位置 $\phi(t)$。在 MATLAB 命令窗口键入：

```
pole(Pointer)
```

将显示：

```
1.0e+002 *
 -9.8734
 -0.0248+0.3104i
 -0.0248-0.3104i
  0.0000
 -0.0871
```

它表明了弹性指针系统有5个极点，其中3个极点在实轴上：一个在原点处；另一个由电机引起，在 -1000 rad/s 附近；还有一个由转子动力特性引起，在 -8.7 rad/s 处。弹性附件增加了一对复数极点 $-2.5\pm31.0i$。如前所述，超前控制器可以提高瞬态响应，但阻尼极点的存在将不利于这种方法的使用。

为了说明超前控制在系统中的局限性，有必要画出比例和超前控制系统的根轨迹。把超前零点放在 -6 正好在第1个稳定极点的右侧。超前极点放在 -50 几乎是零点位置的10倍。则传递函数为：

$$H(s)=\frac{s+6}{s+50}$$

程序为：

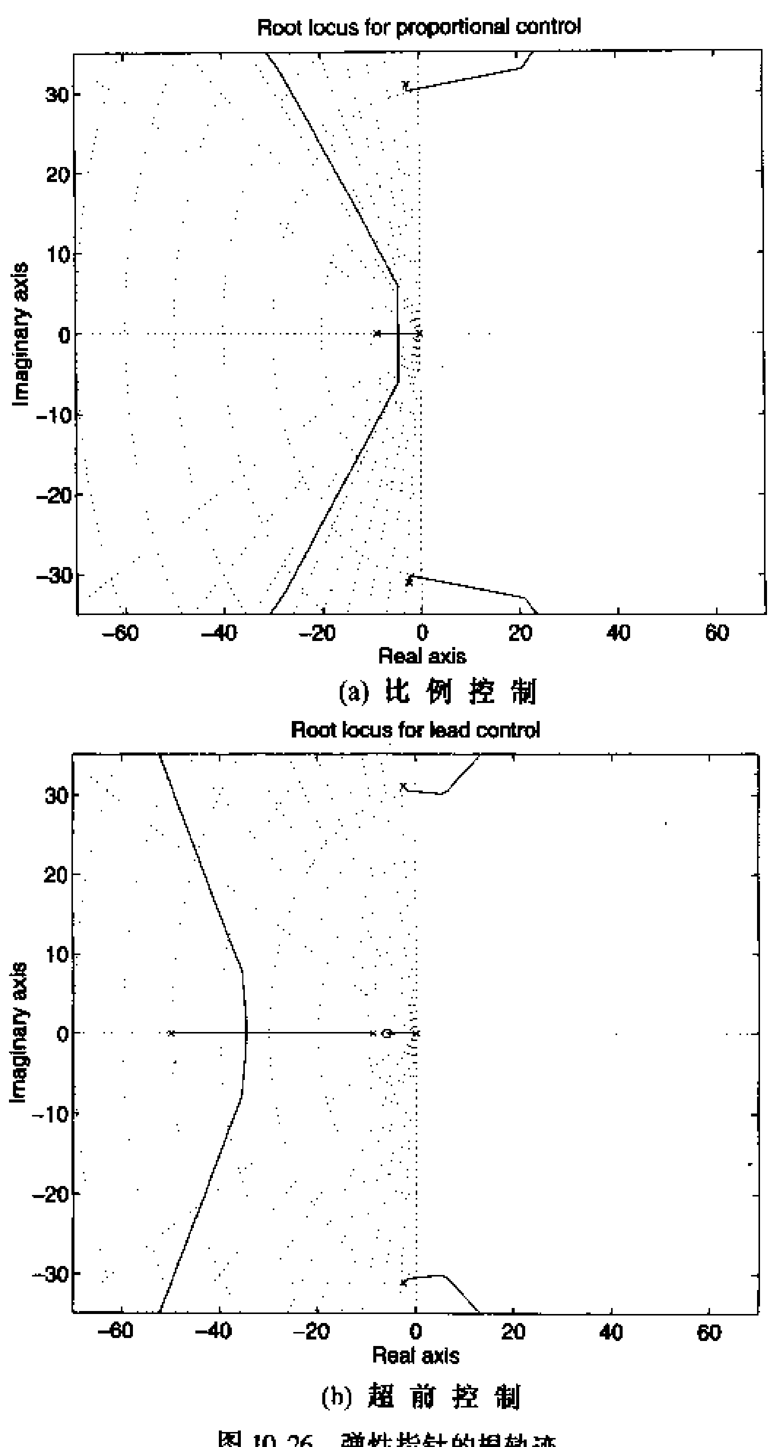
```
rlocus(Pointer);
axis(70*[-1 1 -0.5 0.5]);
sgrid;
xlabel('Real axis');
ylabel('Imaginary axis');
title('Root locus for proportional control');
figure(2);
rlocus(tf([1 6],[1 50])*Pointer);
axis(70*[-1 1 -0.5 0.5]);
sgrid;
xlabel('Real axis');
ylabel('Imaginary axis');
title('Root locus for lead control');
```

程序执行结果如图10.26所示。对于直流电机系统，两种设计方法所允许的增益不是由开环极点的实部所限制，而是由弹性轴引起的复数极点限制。超前控制器的性能与比例控制器的性能相比略强一些。在 MATLAB 命令窗口中输入：

```
rlocus(tf([1 6],[1 50])*Pointer)
```

```
[k,p] = rlocfind(tf([1 6],[1 50]) * Pointer)
```

可以找到超前控制系统和比例控制系统的最佳增益。十字线必须放在由弹性轴引起的复数极点上,因为它是限制增益的主要原因。增益为3时,对于每种设计方法系统都是稳定的,并且具有较好的性能,其闭环系统阶跃响应曲线如图10.28所示。因为标准控制器无意中会产生自由振荡,所以无阻尼复数极点进一步限制了性能的改善。



(a) 比 例 控 制

(b) 超 前 控 制

图 10.26 弹性指针的根轨迹

不产生自由振荡是进一步提高闭环系统性能的关键问题。回到第10.3.2节可以看出,如果零点附近正好有一个极点,那么就不容易产生自由振荡。这里选用超前-滞后控制器,其零点选在 $-3\pm30i$ 处,在自由振荡位置附近。为保证传递函数的正确性,在 -60 位置选择两个

超前－滞后极点，而试图直接消除极点是不现实的。为达到设计要求必须要从一系列的实验数据中获得一个良好的模型。

下面的程序绘出了超前－滞后控制系统的根轨迹。

```
Notch = zpk([-3+30i, -3-30i],[-60, -60],1);
rlocus(tf[1 6],[1 50]) * Notch * Pointer);
axis(70*[-1 1 -0.5 0.5]);
sgrid;
xlabel('Real axis');
ylabel('Imaginary axis');
title('Root locus for notch control');
```

程序执行结果如图 10.27 所示。零点与产生自激的极点非常接近，且系统的根轨迹与没有自激时的根轨迹非常类似。

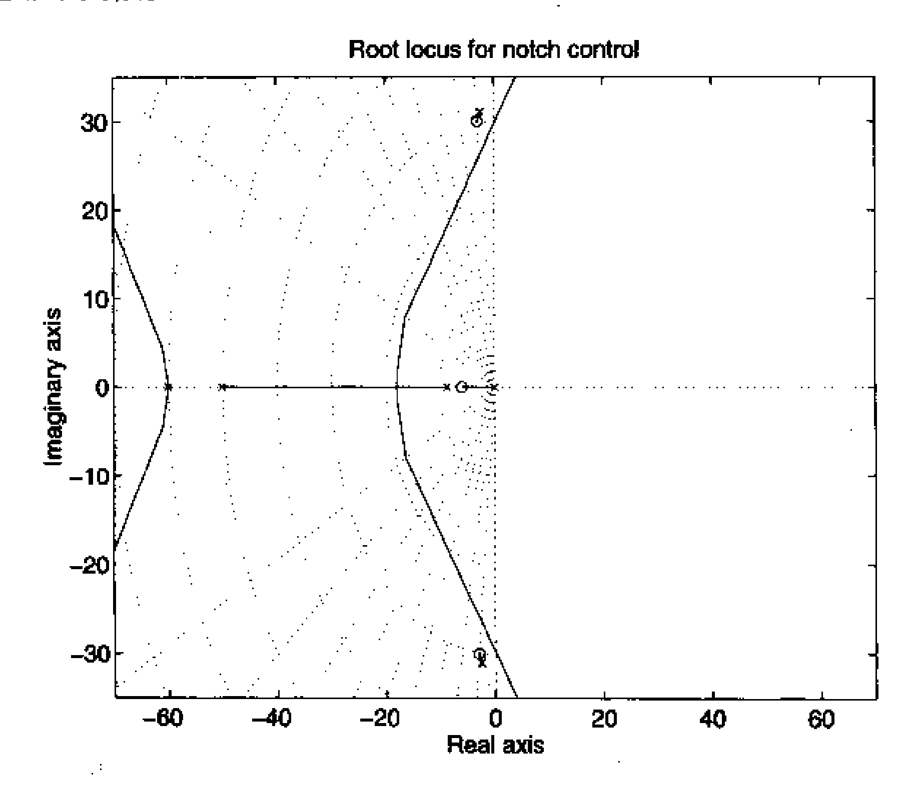


图 10.27 超前－滞后超前补偿的弹性指针的根轨迹

用：

```
rlocus(tf([1,6],[1,50]) * Notch * Pointer)
rlocfind(tf([1,6],[1,50]) * Notch * Pointer)
```

可找到产生一个稳定闭环系统的增益，值为 40.0。十字线放在根轨迹与虚轴的交叉点上，把它们放在以负实轴为中心的 120°楔形内，将得到很好的暂态特性。

下面的程序用于计算前面所讲述的 3 种控制方案的阶跃响应。

```
Lead = tf([1 6],[1 50]);
Notch = zpk([-3+30i, -3-30i],[-60, -60],1);
t = linspace(0,3,200);
yp = step(feedback(3.0 * Pointer,1),t);
yl = step(feedback(3.0 * Lead * Pointer,1),t);
yn = step(feedback(40.0 * Notch * Lead * Pointer,1),t);
```

```
plot(t,yp,'k--',t,yl,'k-.',t,yn,'k-');
legend('Proportional','Lead','Notch');
xlabel('Time');
ylabel('Step response');
title('Comparison of proportional,lead,and notch control');
```

程序执行结果如图 10.28 所示。超前－滞后控制器具有同比例控制器相同的上升时间，但却没有引起轴的自由振荡。

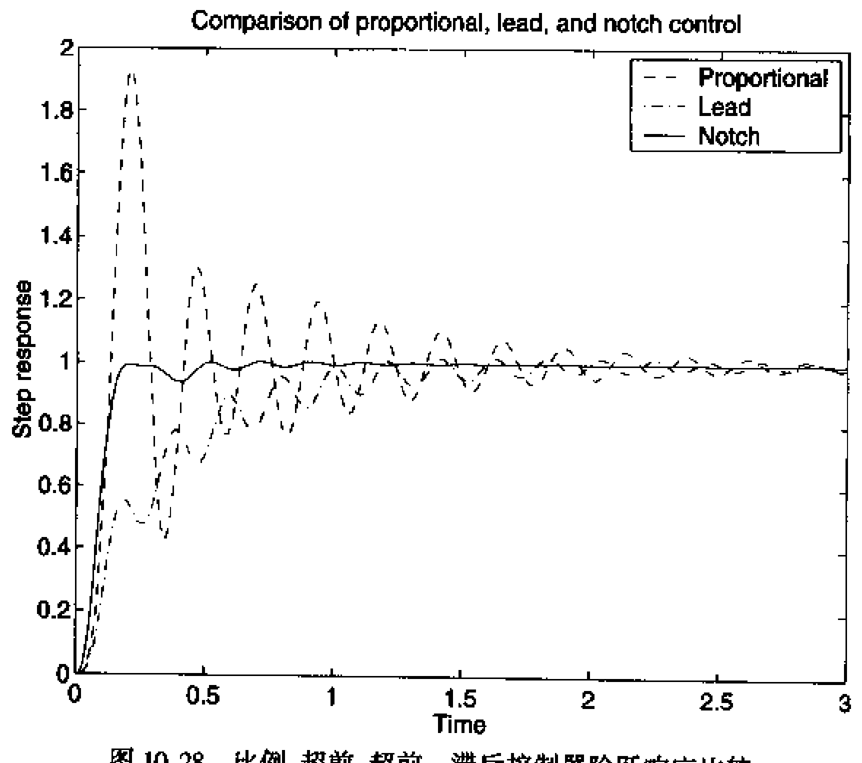


图 10.28 比例、超前、超前－滞后控制器阶跃响应比较

前面的设计只是构成了模拟的控制器，而最终的控制器应该是嵌入式的、廉价的小型计算机。嵌入式控制器用光学编码器周期性地测量指针的位置并与期望位置相比较。通过计算，由数模转换器或脉冲宽度调制器可设置有效的放大器电压。在下面的例子中，微机读取编码器的编码结果，并以每秒一百次的速度更新电压放大器的输出。由于采样时间间隔很小，因此应尽量减小采样间隔内的计算时间。下面的程序设计了对象控制器的离散化模型，并将数字化设计与前面的连续超前－滞后设计进行了比较。数字超前－滞后滤波器的零点设置在 $0.92\pm0.3i$ 处，与对象离散化模型的极点相接近，极点设置为 0.6。数字化传递函数的实现不是绝对固定的。数字超前控制器的零点设置为 0.95，刚好在离散化对象第一个稳定极点 0.5 的右侧。利用：

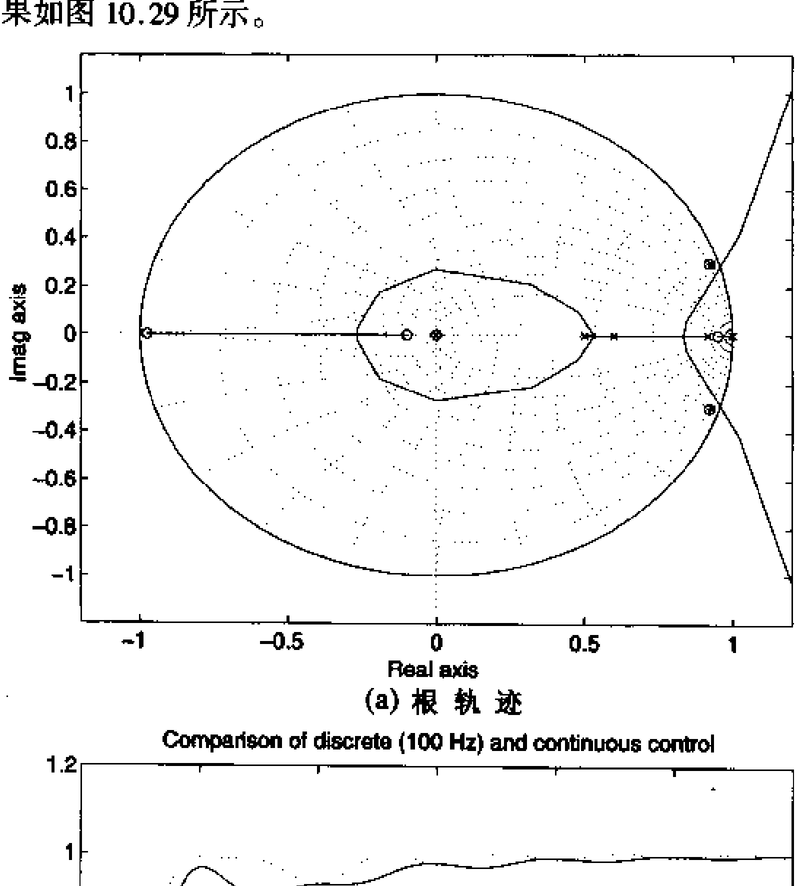
```
rlocus(DNotch * DLead * c2d(Pointer,Ts))
rlocfind(DNotch * DLead * c2d(Pointer,Ts))
```

将十字线放在远离单位圆的根轨迹上，可得出增益 15。

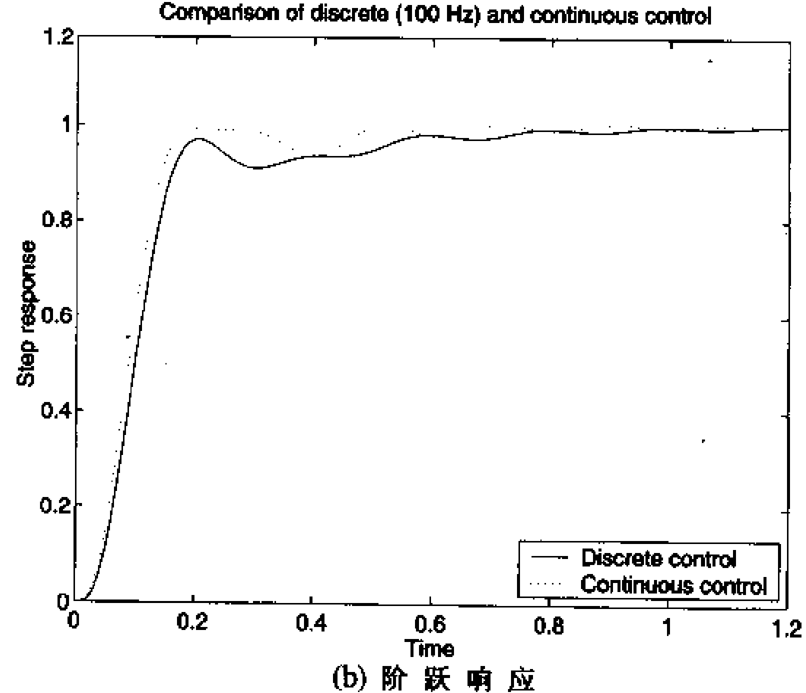
```
Ts = 1/100;
DNotch = zpk([0.92 + 0.3i,0.92 - 0.3i],[0.6 0.6],1,Ts);
DLead = tf([1 -0.95],[1 -0.5],Ts);
rlocus(DNotch * DLead * c2d(Pointer,TS));
axis(1.2 * [-1 1 -1 1]);
zgrid;
```

```
figure(2);
[yd,t] = step(feedback(15 * DNotch * DLead * c2d(Pointer,Ts),1));
Lead = tf([1 6],[1 50]);
Notch = zpk([ - 3 + 30i,-3-30i],[ - 60, - 60],1);
yn = step(feedback(40.0 * Notch * Lead * Pointer,1),t);
plot(t,yd,'k-',t,yn,'k:');
legend('Discrete control','Continuous control',4);
title('Comparison of discrete(100 Hz) and continuous control');
xlabel('Time');
ylabel('Step response');
axis(1.2 * [0 1 0 1]);
```

程序执行结果如图 10.29 所示。



(a) 根 轨 迹



(b) 阶 跃 响 应

图 10.29 弹性指示器系统的数字实现

10.5.2 磁悬浮系统的 PID 控制

研究图 10.30 所示的磁悬浮系统①。其目的是让悬浮的球在受到外界干扰时仍然在期望的高度。球的高度为 $h(t)$，线圈电流为 $i(t)$。磁悬浮运动方程为：

$$m\frac{\mathrm{d}^2h(t)}{\mathrm{d}t^2}=mg-k\left(\frac{i(t)}{h(t)}\right)^2 \tag{10.10}$$

$$L\frac{\mathrm{d}i(t)}{\mathrm{d}t}=v(t)-Ri(t)$$

其中 m 为球的质量，g 为重力加速度，L 为线圈电感，R 为线圈电阻，k 为磁场和球之间的耦合系数。系统输入为线圈电压 $v(t)$，输出为球的高度 $h(t)$。方程为非线性的。

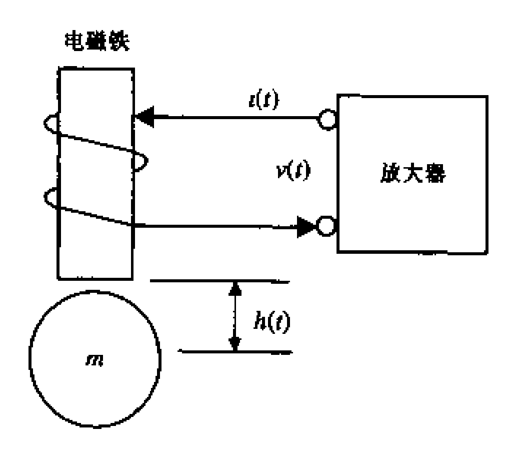


图 10.30 磁悬浮系统

磁场越强，球与电磁体的距离越近，这将破坏系统的稳定性。理想情况下，球与磁铁间应保持足够的距离以使磁力抵消地球的引力。如果球下降得太多，则磁场的作用会变弱而使球完全掉下去。如果球离磁铁太近，则磁场作用又会太强而将球拉向磁铁。因此，首先要计算出地球引力和磁场引力相等的点，这个点称为平衡点。给定期望位置 h_0，可通过令加速度等于零的方法，求出维持该位置的期望电流。即：

$$i_0^2=\frac{mg}{k}h_0^2$$

注意，模型可以在平衡点附近线性化，对式(10.10)线性化可产生一组线性方程。

引入状态变量：

$$x_1=h \quad x_2=\frac{\mathrm{d}h}{\mathrm{d}t} \quad x_3=i$$

则式(10.10)变成：

$$\frac{\mathrm{d}x_1}{\mathrm{d}t}=x_2$$

$$\frac{\mathrm{d}x_2}{\mathrm{d}t}=g-\frac{k}{m}\left(\frac{x_3}{x_1}\right)^2 \tag{10.11}$$

$$\frac{\mathrm{d}x_3}{\mathrm{d}t}=\frac{v}{L}-\frac{R}{L}x_3$$

① B. Friedland, *Advanced Control System Design*, Prentice Hall, Englewood Cliffs, NJ, 1996 年。

方程(10.11)可以在工作点 $x_3 = i_0, x_1 = h_0$ 处通过泰勒级数展开进行线性化,线性化将导致 dx_2/dt 的更改,线性化结果为:

$$\begin{bmatrix} dx_1/dt \\ dx_2/dt \\ dx_3/dt \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{2k}{m}\frac{i_0^2}{h_0^3} & 0 & \frac{-2k}{m}\frac{i_0}{h_0^2} \\ 0 & 0 & -R/L \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + v \begin{bmatrix} 0 \\ 0 \\ 1/L \end{bmatrix}$$

假设球的质量为 100 gm,线圈电阻为 5 Ω,线圈电感为 40 mH,耦合系数为 0.01 Nm²/A,期望高度为 2 cm。首先建立函数 MagLev 描述系统状态空间模型。

```
function Plant = MagLev
m = 0.1;g = 9.82;R = 5;
L = 0.04;k = 0.01;h0 = 0.02;
i0 = h0 * sqrt(m * g/k);
A = [0 1 0;2 * k * i0^2/(m * h0^3)0 - 2 * k * i0/(m * h0^2);0 0  - R/L];
B = [0;0;1/L];
C = [1 0 0];
D = 0;
Plant = ss(A,B,C,D)
```

在 MATLAB 命令窗口键入:

```
MagPoles = pole(MagLev)
```

得到:

```
MagPoles =
   31.3369
  -31.3369
 -125.0000
```

因此,线性化后悬浮系统的极点为 ±31.3,而放大器的极点为 -125。故需要加入一个 PD 控制器来稳定系统。理论上 PD 控制器的传递函数由下式给出:

$$C_0(s) = k_p + sk_d$$

其中 k_p 为比例增益,k_d 为微分增益。控制器的输出是对输入的微分,由于高频噪声的影响这在实际中是很难实现的。一般来说,微分作用可以近似并可通过滤波的方法消除噪声的影响。其结果为:

$$C_1(s) = k_p + k_d \frac{s}{\tau_f s + 1} = k_p \frac{(\tau_f + k_d/k_p)s + 1}{\tau_f s + 1}$$

传递函数 $C_1(s)$ 等价于一个超前控制器,它有时间常数为 $\tau_f + k_d/k_p$ 的零点和时间常数为 τ_f 的极点(滤波器)。之所以选用超前控制器是因为零点总是比极点慢。选择控制器零点为 -20,刚好在磁悬浮系统第一个稳定极点的右侧,滤波器极点为 -50,则 $\tau_f = 20$ ms。为保证系统在正反馈时稳定,在控制器中要改变符号,传递函数为:

$$C_1(s) = -\frac{s + 20}{s + 50}$$

程序为:

```
PD=tf(-1*[1 20],[1 50]);
rlocus(PD*MagLev);
sgrid
title('Root locus of PD controlled magnetic levitator');
xlabel('Real axis');
ylabel('Imaginary axis');
```

程序执行结果的根轨迹如图 10.31 所示。

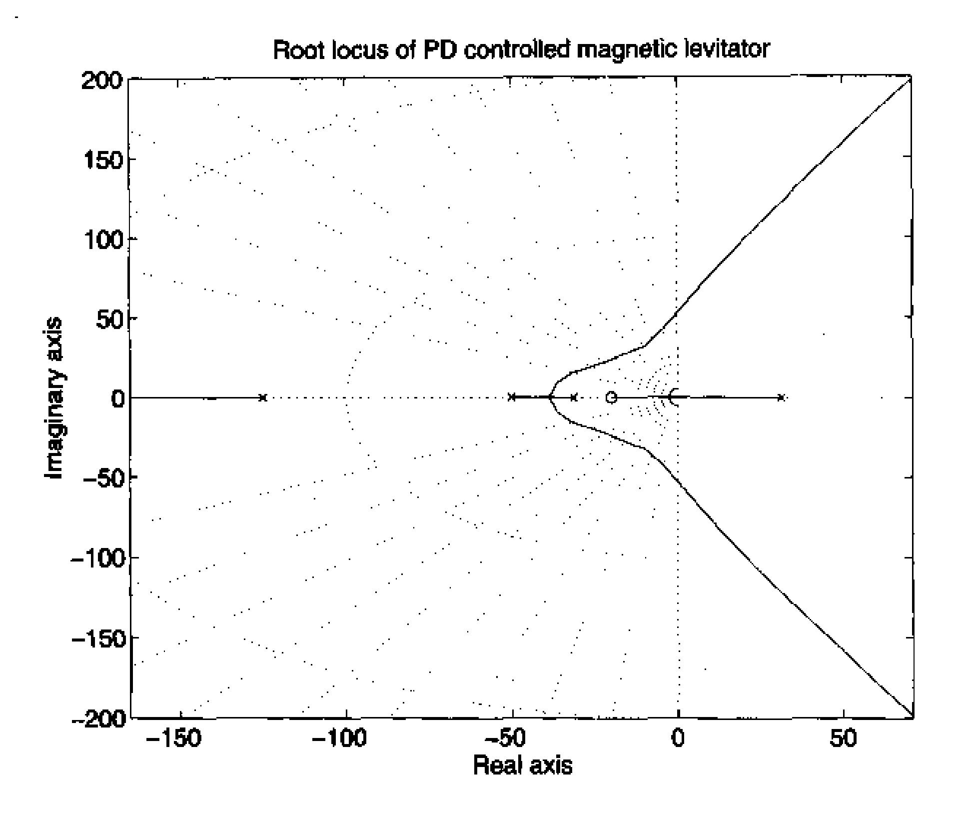


图 10.31 PD 串联控制器的磁悬浮系统根轨迹

从图中可以看出,在某些低增益情况下,不稳定极点被移到了左半平面;而在高增益时,复数共轭极点则进入了右半平面。在 MATLAB 命令窗口键入:

```
rlocus(tf(-1*[1 20],[1 50])*MagLev)
rlocfind(tf(-1*[1 20],[1 50])*MagLev)
```

并将十字线放在实轴根轨迹上,介于不稳定极点与控制器零点之间,则可由图中得到一个稳定的增益值。在虚轴和控制器零点之间近于一半的位置,产生增益 150。因此 $k_p=-60, k_d=-1.8$,且滤波器时间常数为 20 ms,即相当于 1/50。

在测定 $v_0=Ri_0$ 时,最可能引起测量误差和模型化误差,从而使球的位置 $h(t)$产生稳态误差。PD 影响系统的不稳定性和暂态特性,而 PI 则经常用于改善系统的稳态特性。研究 PI 控制器:

$$C_2(s)=k_p+\frac{k_i}{s}=\frac{k_p s+k_i/k_p}{k_i\quad s}$$

它带有参数 k_p 和 k_i。PI 控制器有一个极点在原点处,一个零点在 $-k_i/k_p$ 处。相对于系统其他零极点而言,如果 PI 控制器零点与极点相距很近,那么,当 PI 控制器与 PD 控制器串联形成 PID 控制器时,它对于闭环暂态特性的影响是可以忽略的。因此,反馈增益 150 仍可以使用。

PI控制器对于稳态误差的影响很大，在该控制系统中选择 $k_p = k_i = 1$。下面的程序仿真了线性磁悬浮模型闭环系统的脉冲响应。

```
PD = tf( - 1 * [1 20],[1 50]);
PI = tf([1 1],[1 0]);
[y,t] = impulse(feedback(150 * PI * PD * MagLev,1));
plot(t,y);
grid;
xlabel('Time');
ylabel('Impulse response');
title('Impulse response of a magnetic levitator');
```

程序执行结果如图10.32所示。结果返回闭环系统的线性化模型，但更为重要的是确定回路中非线性模型系统的稳定性。

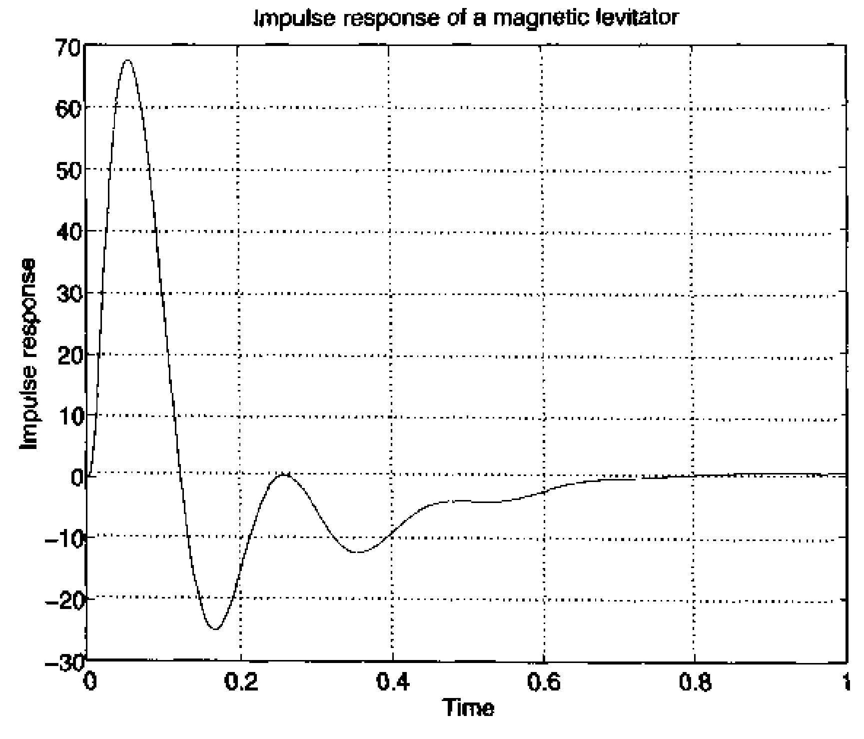


图10.32 磁悬浮系统线性模型的脉冲响应曲线

非线性系统可用SIMULINK进行仿真。为此需要用到一个称为S-Function的模块，该模块是由用户定义的函数。这里将该用户定义的函数称为MagModel。SIMULINK的S-Function模块指定需对MagModel函数按如下顺序传递4个参数：时间 t、状态变量 x、输入量 u 和一个整数 *flag*。离散和连续状态的混合模型可用S-Function建立。仅研究连续非线性化模型部分，SIMULINK查询用户函数以确定非线性模型，*flag* 决定查询的目的。当 *flag* = 1时，函数返回由式(10.11)确定的 x 关于 t 的导数、状态 x 和输入量 u；当 *flag* = 3时，函数返回输出量；当 *flag* = 0时，函数返回向量 *sys*，其元素依次为：连续状态数、离散状态数、输出量数目、输入量数目、根的数目和一个最终标识，当系统有直接反馈时该标识置1。在磁悬浮问题中，*sys* = [3 0 1 1 0 0]，说明有三个连续状态，没有离散状态，一个输入，一个输出，无根且没有直接反馈。当 *flag* = 0时，也返回连续状态的初始条件，即系统第一次运行时的状态。系统的平衡位置为 $x(0) = [h(0)\ 0\ i(0)]'$。启动系统时不在其平衡位置，而是赋予 $h(0)$ 一个比平衡值大10%的初始值。在这样的初始条件下控制器会开始工作，否则球会脱离磁体。因为控制器是基于系统

的线性模型设计的,所以平衡位置的偏差一定要小。

```
function[sys,x0] = MagModel(t,x,u,flag)
m=0.1;g=9.82;R=5;L=0.040;
k=0.01;h0=0.02;
i0=h0*sqrt(m*g/k);
switch flag
 case 1
  xdot=zeros(3,1);
  xdot(1)=x(2);
  xdot(2)=m*g-k*x(3)^2/x(1)^2;
  xdot(3)= -R/L*x(3)+1/L*u(1);
  sys=xdot;
 case 3
  sys=x(1);
 case 0
  sys=[3 0 1 1 0 0];
  x0=[h0+0.1*h0;0;i0];
 otherwise
  sys=[];
end
```

控制器由变量名而不是由数值确定,因此,这些变量在 SIMULINK 运行前必须要在 MATLAB 命令窗口中定义。运行下面的程序可生成这些变量,并运行 SIMULINK:

```
PD=tf(-1*[1 20],[1 50]);
PI=tf([1 1],[1 0]);
v0=0.991;h0=0.02;
[num,den]=tfdata(150*PD*PI,'v');
simulink;
```

$v0$ 和 $h0$ 的值由给定值算出。

用 MATLAB 建立如图 10.33 所示的方框图。从 Math 组件库选用 Sum 和 Gain 模块。从 Function & Tables 组件库选用 S-Function 模块,双击模型窗口的 S-Function 模块并用 MagModel 作为 S-Function 模块的名称。

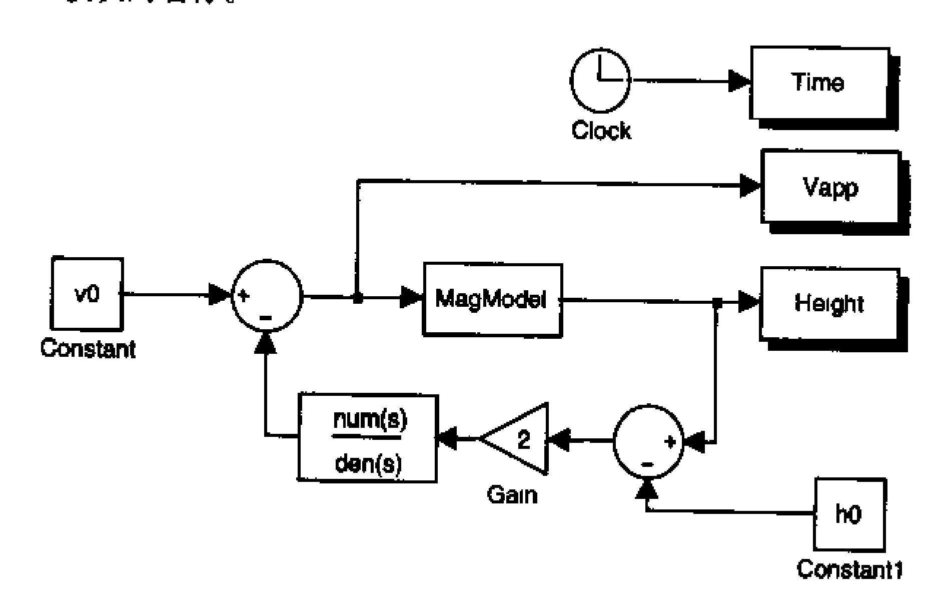


图 10.33 非线性磁悬浮系统的 SIMULINK 方框图

从 Continuous 组件库选用 Transfer Fnc 模块，双击此模块，并分别输入分子和分母的变量名，本例中为 *num* 和 *den*。在进入 SIMULINK 前，在程序的第 3 行对这两个数进行了赋值。

从 Source 组件库中选用 Constant 模块，并输入电压偏差和高度偏差，这些值用来描述误差。如果变量名已经或即将在 MATLAB 命令窗口中赋值，可通过双击模块输入它们的数值或输入它们的变量名。因为 $v0$ 和 $h0$ 在前面运行的程序中已经定义，所以选用后一种方法。

最后，从 Sink 组件库中选用 To Workspace，记录作为时间函数的承载高度（*Height*）和磁铁线圈电压（*Vapp*）的幅值。时间来自 Source 组件库的 *Clock*，时间值存储在 *Time* 数组中。三个模块周围的阴影是通过 Format 下拉菜单的 Show Drop Shadow 命令得到的。在每一个仿真时刻，*Vapp* 和 *Height* 的值保存在工作空间里，并可以通过函数 Plot 来显示。图 10.34 显示了 *Vapp* 和 *Height* 随时间变化的曲线，它们是通过在 MATLAB 命令窗口中运行下列程序得到的。*Height* 的值进行了 50 倍放大。

```
plot(Time,50 * Height,'k-',Time,Vapp,'k- ')
legend('50h(t)','v(t)')
text(1,1.5,'Initial conditions')
text(1.2,1.45,'h(0) = 0.022m')
text(1.2,1.4,'v(0) = 0.991 V')
xlabel('Time(s)')
ylabel('V_{app} and h(t)')
axis([0 4.7 1.6])
```

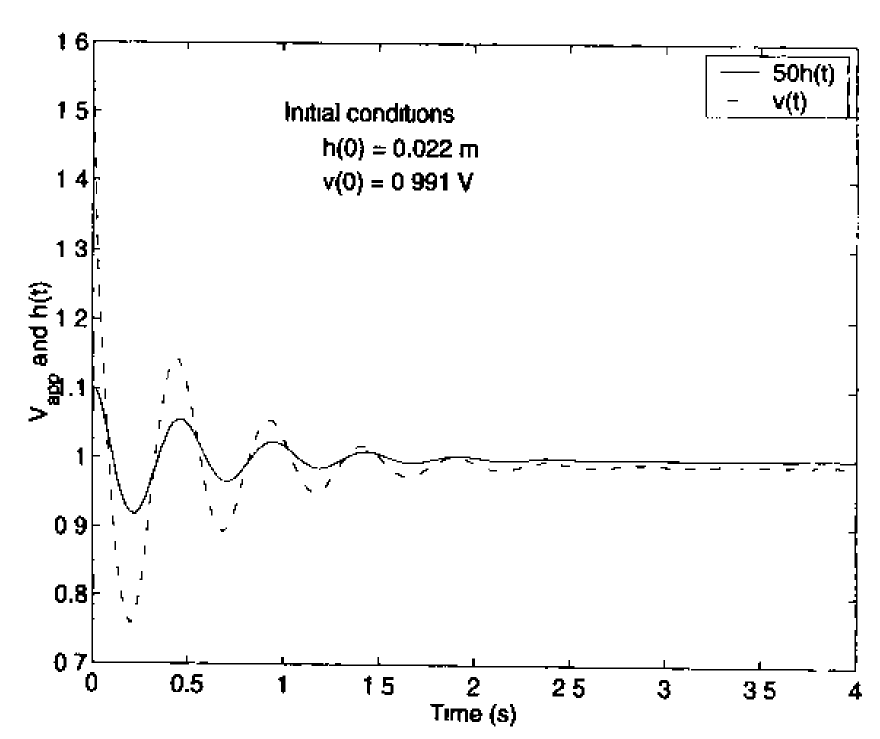


图 10.34　带有初始条件和模型误差的非线性磁悬浮系统在 PID 控制下的响应曲线

10.5.3　倒置摆的超前控制

用根轨迹方法可为倒置摆设计超前控制器。研究图 10.35 所示的装在圆盘上的倒置摆。控制系统的目标是通过控制圆盘的位置保持摆的垂直。圆盘的角度 Ψ 和摆的角度 θ 都是可以测量的，其运动方程为：

$$ml^2\frac{d^2\theta}{dt^2}+mrl\cos(\theta)\frac{d^2\Psi}{dt^2}=mgl\sin(\theta)+b_1\frac{d\theta}{dt}$$

$$mrl\cos(\theta)\frac{d^2\theta}{dt^2}+(J+mr^2)\frac{d^2\Psi}{dt^2}=mrl\sin(\theta)\left(\frac{d\theta}{dt}\right)^2+b_2\frac{d\Psi}{dt}+\tau_m \tag{10.12}$$

其中 m 为摆锤的质量，l 是摆的长度，r 是圆盘的半径(也可以是摆锤连接部分的半径)，d 是圆盘的厚度，$J=\rho\pi dr^4/4$ 为圆盘的惯量，b_1 是摆旋转的摩擦系数，b_2 是圆盘旋转的摩擦系数，τ_m 是与圆盘底部相连的电动机的转力矩。与磁性轴承类似，当 θ 和 $d\theta/dt$ 很小时在工作点附近运动方程可以线性化。

定义状态向量 x 为：

$$x_1(t)=\theta(t)\quad x_3(t)=\frac{d\theta}{dt}$$

$$x_2(t)=\Psi(t)\quad x_4(t)=\frac{d\Psi}{dt}$$

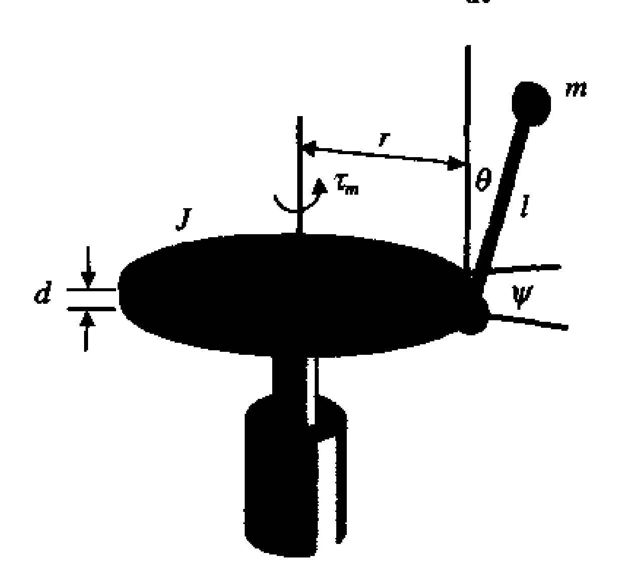


图 10.35 圆盘上的倒置摆

将这些方程代入式(10.12)，并假设 θ 和 $d\theta/dt$ 非常小，可以忽略掉2阶以上的项，得到：

$$\frac{dx_1}{dt}=x_3$$

$$\frac{dx_2}{dt}=x_4$$

$$ml^2\frac{dx_3}{dt}+mlr\frac{dx_4}{dt}=mglx_1+b_1x_3$$

$$mlr\frac{dx_3}{dt}+(J+mr^2)\frac{dx_4}{dt}=b_2x_4+\tau_m$$

或矩阵形式：

$$M\dot{x}=Qx+Wu$$

其中，

$$W=[0\ 0\ 0\ 1]'\quad x=[x_1\quad x_2\quad x_3\quad x_4]'\quad \dot{x}=dx/dt$$

$$M=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & ml^2 & mlr\\ 0 & 0 & mlr & J+mr^2\end{bmatrix}\quad Q=\begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ mgl & 0 & b_1 & 0\\ 0 & 0 & 0 & b_2\end{bmatrix}\quad u=\tau_m$$

对应于 $x(t)=0$ 的垂直位置是系统在 $u=\tau_m=0$ 时的一个平衡点。在垂直位置倒置摆系统的线性化方程可以用函数 Pendulum 描述状态空间。系统的输出为摆角 θ 和圆盘角度 Ψ。因此,摆的输出对象是一个单输入双输出的系统,它可以用下面的方法描述。Plant(1,1)是从 τm 到 θ 的传递函数,而 Plant(2,1)是从 τ_m 到 Ψ 的传递函数。假设摆的长度为 30 cm,摆锤的质量为 0.2 kg,圆盘的半径为 15 cm,圆盘的厚度为 1 cm,且其密度为 2 500 kg/m^3。系统的摩擦系数设为 0,即 $b_1=b_2=0$。

```
function Plant = Pendulum
l = 0.3;g = 9.81;m = 0.2;r = 0.15;
d = 0.01;rho = 2500;b1 = 0;b2 = 0;
J = 0.25 * pi * rho * d * r^4;
M = [1 0 0 0;0 1 0 0;0 0 m * l^2 m * r * l;0 0 m * r * l J + m * r^2];
Q = [0 0 1 0;0 0 0 1;m * g * l 0 b1 0;0 0 0 b2];
W = [0;0;0;1];
A = inv(M) * Q;
B = inv(M) * W;
C = [1 0 0 0;0 1 0 0];
D = [0];
Plant = ss(A,B,C,D);
```

参照例 5.2。

通过在 MATLAB 命令窗口中键入:

```
pole(Pendulum)
```

得到系统极点。显示:

```
0
0
6.8923
-6.8923
```

有两个极点在原点处,另外两个在实轴 ±6.9 rad/s 处。对称于虚轴有一对极点,这表明系统在开环状态下是不稳定的。在 MATLAB 命令窗口中输入:

```
plant = Pendulum;
tzero(Plant(1,1))
```

结果显示两个零点,也就是从摆的角度 θ 观测到的传输零点。只有第一个输入 τ_m 和第一个输出 θ 的系统可以用矩阵的形式访问,即 Plant 的下标(1,1)。这样就有两个零点在极点的正上方。这说明仅根据输出 θ 控制倒置摆会忽略某些动态特性,尤其当摆角未包含足够的信息来表示盘的位置和速度时,这种情况更加明显。因此当圆盘以恒定的角速度旋转时,角度传感器的输出为零。尽管输出为零,但动态特性仍然存在,有时称之为零动态。这些不能观测的零动态不能仅通过 θ 的反馈进行稳定。从圆盘角度 Ψ 观测的摆的零点可通过在 MATLAB 命令

窗口中输入：

```
Plant = Pendulum;
tzero(Plant(2,1))
```

获得。由于对象有两个输出，因此可用下标（2，1）获得子系统的第 2 个输出和第 1 个输入。从这些结果中可以看出，零点位于 ±5.7，与摆的开环极点 ±6.9 接近。这说明从盘的位置很难看出摆的特性。根据任何一个输出设计而成的单入单出（SISO）控制器，其运行特性都很差。

依赖于两个输出的多入单出(MISO,multiple-input single-output)控制器,其运行特性要比 SISO 控制器好很多。这将是进一步设计的关键所在。设计图 10.36 所示的控制器可分为两步：首先对输出 θ 设计控制器,保持摆的垂直,然后围绕这个控制器对 Ψ 设计控制器,保持圆盘在指定位置。

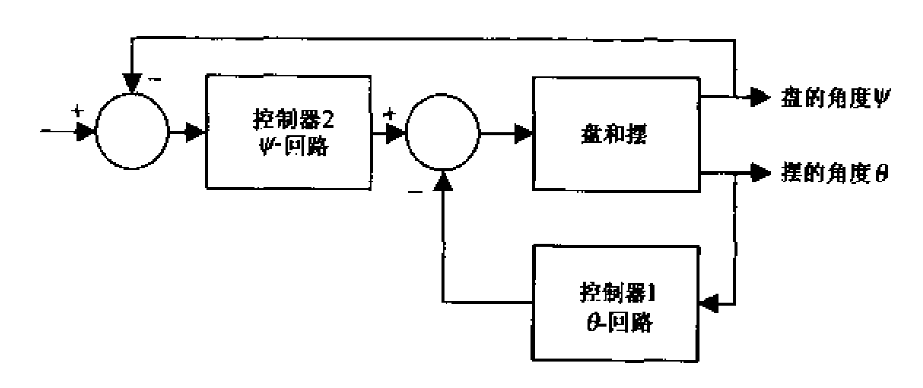


图 10.36　倒置摆控制系统的方框图

下面用超前控制器保持摆的垂直,这样的控制器用输出 θ 来确定采用哪种补偿方式。参照位于 ±6.9 处的开环极点,将超前零点放在 −5 处,正好位于稳定极点的右侧,其目的是将不稳定极点移到复平面的左半部分。超前极点位于 −10,因此,

$$C_\theta(s) = -\frac{s+5}{s+10}$$

由于要求为正反馈,故此 $C_\theta(s)$中出现负号。

下面的程序用于生成 θ 控制回路的根轨迹图。MATLAB 可自动选择增益范围,但该例中采用在 $0.1 \leqslant \theta \leqslant 10$ 的范围内按对数比例进行等分。

```
Plant = Pendulum;
PlantTheta = minreal(Plant(1,1));
ControlTheta = tf( - 1 * [1 5],[1 10]);
rlocus(ControlTheta * PlantTheta, logspace( - 1,1,60));
sgrid;
xlabel('Real axis');
ylabel('Imaginary axis');
title('Root locus of \ theta control loop');
```

程序执行结果如图 10.37(a)所示。

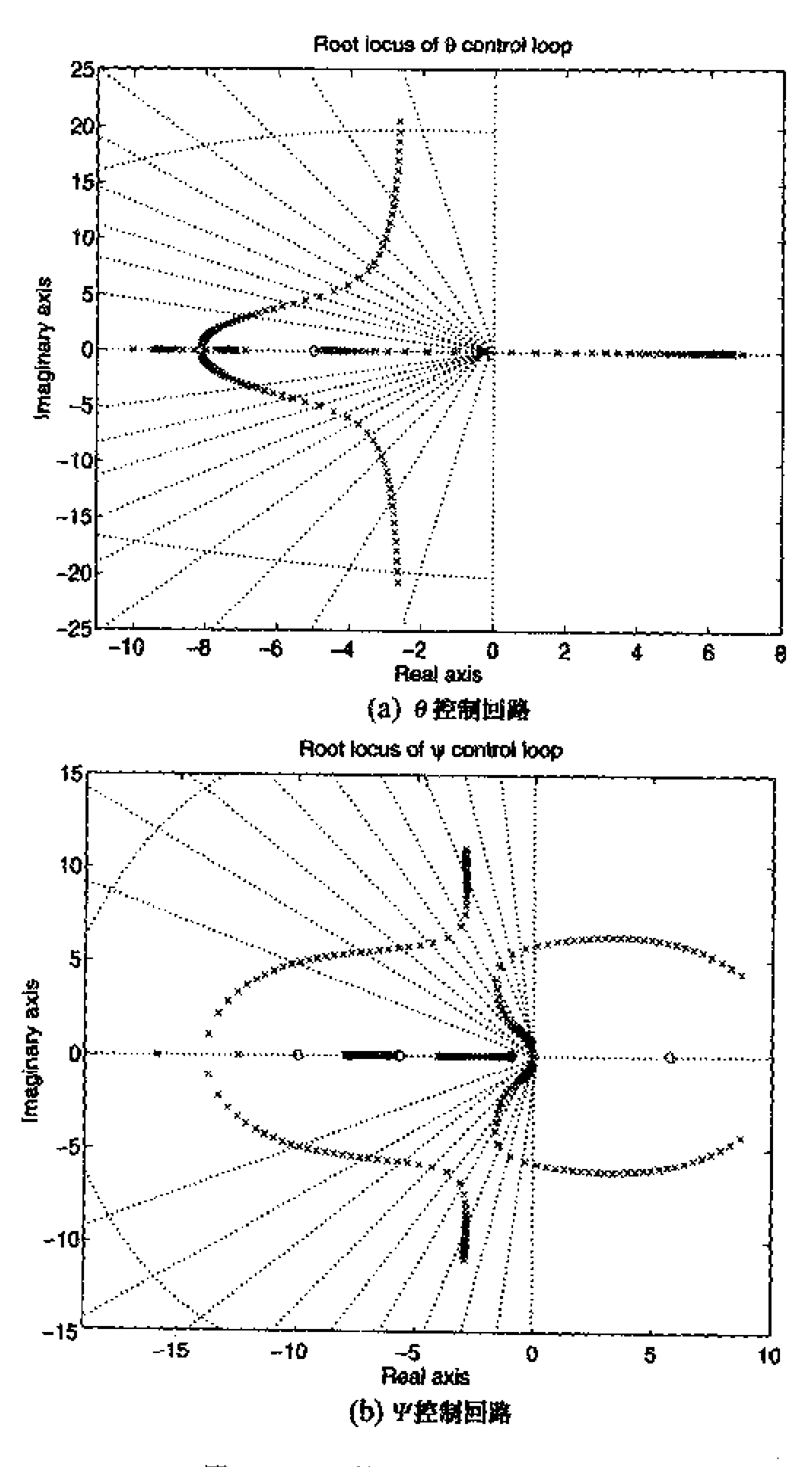
增益可通过下列程序获得：

```
rlocus(ControlTheta * PlantTheta)
```

```
rlocfind(ControlTheta * PlantTheta)
```

将十字线放在虚轴与超前零点之间的实轴上，则可得到使 θ 回路稳定的增益。由下面的程序可得出 θ 控制系统的闭环极点。

```
Plant = Pendulum;
PlantTheta = minreal(Plant(1,1));
ControlTheta = tf(-1*[1 5],[1 10]);
pole(feedback(4 * ControlTheta * PlantTheta,1))
```



(a) θ 控制回路

(b) Ψ控制回路

图 10.37　反馈控制回路的根轨迹

程序执行结果显示,闭环系统极点近似为 -4 和 -3±11i,系统尽管无阻尼,但是稳定的。

现在设计圆盘位置 Ψ 的外部反馈回路。外部反馈回路与内部反馈回路相关,因此首先将第一个控制器置入 θ 回路中形成闭环回路系统。可通过下列程序实现该项操作。

```
ControlTheta = tf( - 1 * [1 5],[1 10]);
PlanPsi = feedback(Pendulum,4 * ControlTheta,[1],[1]);
pole(PlantPsi)
tzero(PlantPsi)
```

注意,因为对象是 MIMO,而控制器是 SISO,所以必须指定使用哪个输入和哪个输出。结果显示 *PlantPsi* 在原点处还有两个附加的极点,相当于一个双积分器。这就需要将它们向左移,其目的是产生一个不稳定的零点。倒置摆是非最小相位系统的例子。为了转动圆盘,控制器必须先向相反的方向转动,以保持摆从当前位置运动到期望位置时一直保持垂直。不稳定零点可以吸收双积分器中的一个极点。为了解决这个问题再次使用超前控制器,其零点刚好在复平面的左半平面。下面的程序画出了 Ψ 控制器回路的根轨迹。

```
ControlTheta = tf( - 1) * [1 5],[1 10]);
PlantPsi = feedback(Pendulum,4 * ControlTheta,[1],[1]);
ControlPsi = tf( - 1 * [1 1],[1 8]);
k = 0.35 * logspace( - 1,1,60);
rlocus(ControlPsi * PlantPsi(2,1),k);
sgrid;
xlabel('Real axis');
ylabel('Imaginary axis');
title('Root locus of \ psi control loop');
```

程序执行结果如图 10.37(b)所示。在 MATLAB 命令窗口键入:

```
rlocus(ControlPsi * PlantPsi)
rlocfind(ControlPsi * PlantPsi)
```

在靠近回路进入复平面的根轨迹线的下半部分放置十字线可得到合适的增益。增益值为 0.3 时,将所有极点放置在左半平面。

最后控制系统的阶跃响应曲线可由下面的程序算出:

```
ControlTheta = tf( - 1 * [1 5],[1 10]);
PlantPsi = feedback(Pendulum,4 * ControlTheta,[1],[1]);
ControlPsi = tf( - 1 * [1 1],[1 8]);
[y,t] = step(feedback(0.3 * ControlPsi * PlantPsi,1,[1],[2]));
plot(t,y( : ,1),'k-',t,y( : ,2),'k--');
xlabel('Time');
ylabel('Step response');
title('Response to a step command for pendulum system.');
legend(' \ theta(t)',' \ psi(t)');
```

阶跃响应曲线如图 10.38 所示。结果揭示了控制器和被控对象具有非最小相位系统的特性。

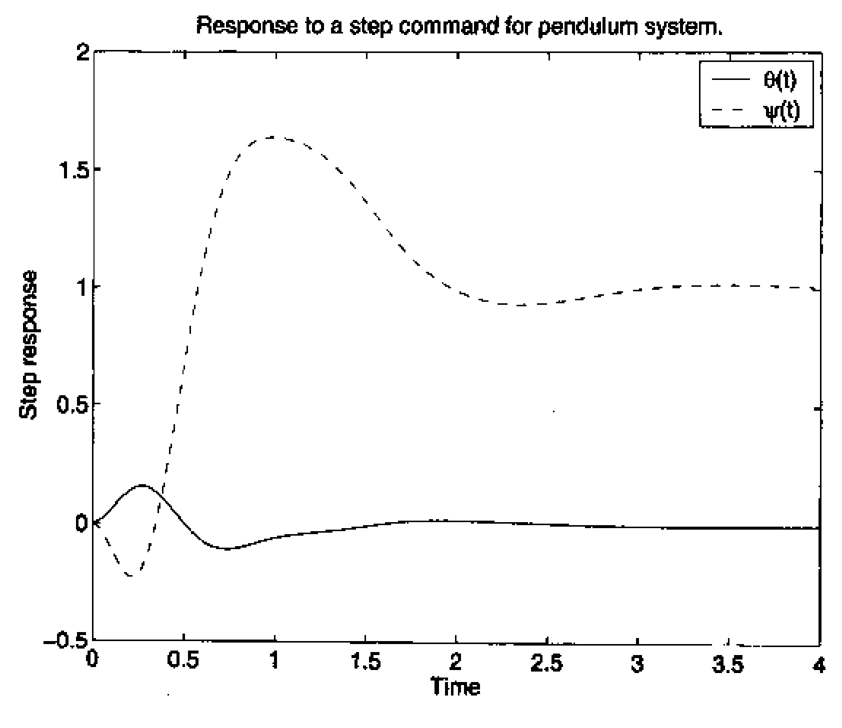


图 10.38 倒置摆的阶跃响应曲线

10.5.4 磁悬浮飞轮的控制

研究图 10.39 所示的磁悬浮飞轮系统。电磁线圈用来使飞轮悬浮,所以飞轮能够以很高的速度运行,而且没有摩擦损耗。控制系统的目标是保持飞轮的悬浮状态。与第 10.5.2 节描述的磁悬浮系统一样,系统实质上是不稳定的。输出为 4 个可以测量的距离。在可测量的惯性参照系中,这些距离对应于轴的顶部和底部 x 和 y 的位置。4 个线圈电流控制轴的磁场,每个线圈都配有传感器。

$$\frac{d^2 x_{cm}}{dt^2} = \frac{f_1 + f_3}{m}$$

$$\frac{d^2 y_{cm}}{dt^2} = \frac{f_2 + f_4}{m}$$

运动的线性方程为:

$$\frac{d^2 \phi}{dt^2} = -\beta\omega \frac{d\Psi}{dt} + \frac{h}{J_{xx}}(f_4 - f_2)$$

$$\frac{d^2 \Psi}{dt^2} = \beta\omega \frac{d\phi}{dt} + \frac{h}{J_{xx}}(f_1 - f_3)$$

其中 x_{cm} 和 y_{cm} 是在可测量的惯性参照系中飞轮的重心位置,在惯性参照系中,(ϕ, Ψ)用旋转(ϕ)和倾角(Ψ)给出飞轮参照系的方位和偏转方向①,m 为飞轮的质量,J_{xx} 是相对于非自旋轴的飞轮的转动惯量,$\beta = J_{zz}/J_{xx}$,h 为重心到调节器的距离。输入 f_i 为磁性轴承的作用力并遵循如下关系式:

① 参见 R. M. Murray, X. Li 和 S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, CRC Press, Boca Raton, FL, 1994 年。

$$f_i = k_1 y_i + k_2 u_i$$

其中 y_i 是从飞轮到调节器的距离,也是系统的输出。轴是由缠有线圈的永磁铁构成的。负恢复常数 k_1 由永磁铁确定,增益 k_2 由线圈中的电流 u_i 产生的磁场确定。轮的运行速度为 ω rad/s。

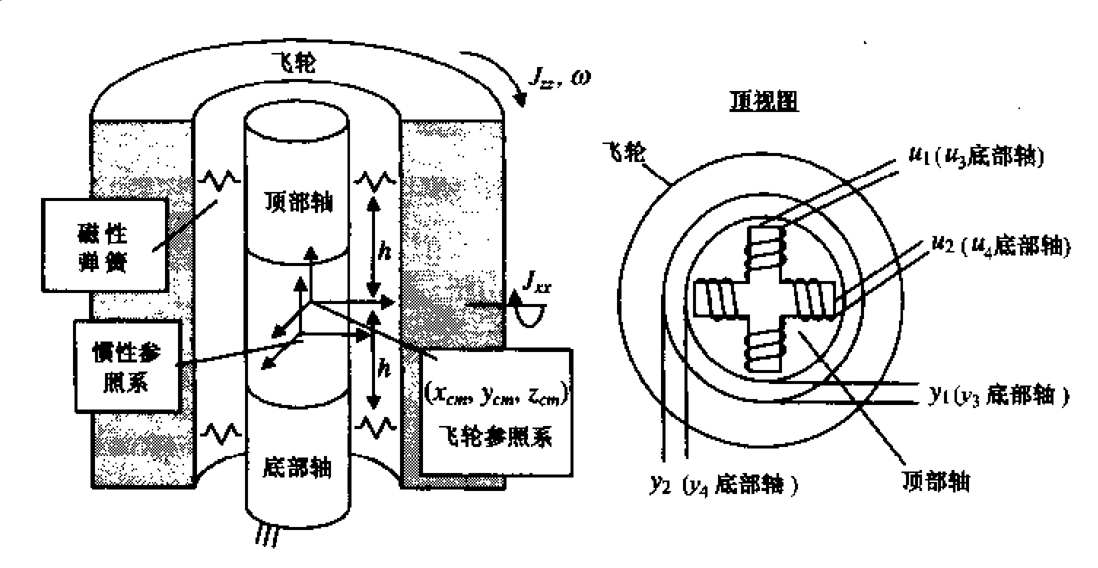


图 10.39 磁悬浮飞轮

对于很小的 ϕ 和 Ψ 值,y_i 由下式给出:

$$y_1 = x_{cm} + h\Psi$$
$$y_2 = y_{cm} - h\phi$$
$$y_3 = x_{cm} + h\Psi$$
$$y_4 = y_{cm} - h\phi$$

如果令:

$$q(t) = [x_{cm}, y_{cm}, \phi(t), \Psi(t)]'$$

且

$$u(t) = [u1\ u2\ u3\ u4]'$$

则线性化方程可以写为:

$$\ddot{q} = \omega \boldsymbol{P}_a \dot{q} + k_1 \boldsymbol{B}_a \boldsymbol{C}_a q + k_2 \boldsymbol{B}_a u$$
$$y = \boldsymbol{C}_a q$$

其中,

$$\boldsymbol{P}_a = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta \\ 0 & 0 & -\beta & 0 \end{bmatrix}$$

$$\boldsymbol{B}_a = \begin{bmatrix} 1/m & 0 & 1/m & 0 \\ 0 & 1/m & 0 & 1/m \\ 0 & -h/J_{xx} & 0 & h/J_{xx} \\ h/J_{xx} & 0 & -h/J_{xx} & 0 \end{bmatrix} \quad (10.13)$$

$$C_a=\begin{bmatrix}1 & 0 & 0 & h\\ 0 & 1 & -h & 0\\ 1 & 0 & 0 & -h\\ 0 & 1 & h & 0\end{bmatrix}$$

若设：

$$z=\begin{bmatrix}q\\ \dot{q}\end{bmatrix}$$

可得到下面的矩阵方程：

$$\dot{z}=\begin{bmatrix}0 & I\\ k_1 B_a C_a & \omega P_a\end{bmatrix}z+\begin{bmatrix}0\\ k_2 B_a\end{bmatrix}u$$

$$y=[C_a \quad 0]z+[0]u$$

其中 I 为单位矩阵。

下面的函数 Flywheel 可生成线性化的模型，它作为旋转速度的函数，转速单位为 r/m。此模型将应用于下面的设计计算中。如果该函数没有输入参数，则假定其运行速度为 0 r/m。对于本例中的飞轮，x 和 y 轴的转动惯量为 1.563×10^{-4} Nm·s^2，z 轴的转动惯量为 1.141×10^{-4} Nm.s^2，无量纲形式的因子 β 近似为 1。飞轮的质量为 340 gm，重心的高度为 3 cm。线圈系数 $k_1=4.8\times10^4$ N/m，$k_2=3.75$ N/A。

```
function Plant = Flywheel(rpm)
if nargin < 1, rpm = 0; end;
Jxx = 1.563e-4;
Jzz = 1.141e-4;
beta = Jzz/Jxx;
m = 0.34;
h = 0.03;
k1 = 4.8e4;
k2 = 3.75;
omega = rpm/60 * 2 * pi;
Pa = [zeros(1,4);zeros(1,4);0 0 0 -beta;0 0 beta 0];
Ba = [1/m 0 1/m 0; 0 1/m 0 1/m;0 -h/Jxx 0 h/Jxx;h/Jxx 0 -h/Jxx 0];
Ca = [1 0 0 h;0 1 -h 0; 1 0 0 -h;0 1 h 0];
A = [zeros(4),eye(4);k1 * Ba * Ca,omega * Pa];
B = [zeros(4);k2 * Ba];
C = [Ca zeros(4)];
D = [zeros(4)];
Plant = ss(A,B,C,D);
```

开环极点随 r/m 变化。运行下面的程序可画出一组在典型运行速度条件下的开环极点。

```
rpm = [0:100:16000 16100:10:20000];
result = zeros(8,length(rpm));
for j = 1:length(rpm);
 result(:,j) = pole(Flywheel(rpm(j)));
end
plot(real(result(:1)),imag(result(:,1)),'x');
hold on;
for j = 1:8;
x = real(result(j,:));
```

```
    y = imag(result(j,:));
    plot(x,y);
  end;
  grid;
  xlabel('Real axis');
  ylabel('Imaginary axis');
  title('RPM root locus of the flywheel');
```

程序执行结果如图 10.40 所示。有两组对称于虚轴的极点(没有零点),一组位于 ±530 rad/s(85 Hz)与运行速度无关,是由直线运动模型而引起的;另一组位于 ±740 rad/s(120 Hz)受陀螺耦合影响,是由两个旋转模型而引起的。

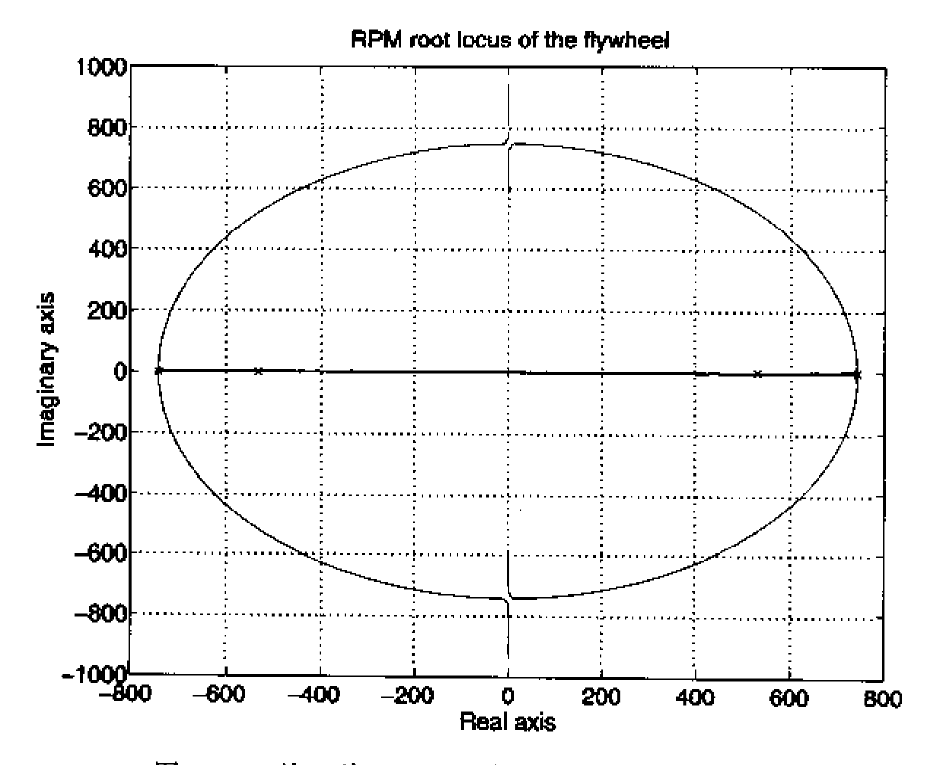


图 10.40 从 0 到 20 000 r/m 的运行速度函数的根轨迹

总之,有 2 个不稳定极点和 2 个稳定极点,有一半的极点随运行速度改变。察看式(10.13)可以看出,在 $\omega=0$ r/m 处,$\boldsymbol{P}$ 不影响结果。这说明飞轮在转速为 0 rad/m 时可实现静态解耦,并形成两个旋转、两个平移的 SISO 系统。因此,像根轨迹法这样的传统超前控制方法可用来设计稳定的控制器。下面来研究矩阵 $\boldsymbol{T}$ 在多个输入通道之间形成信号的和与差。

$$\boldsymbol{T}=\begin{bmatrix}1 & 0 & 0 & 1\\0 & 1 & -1 & 0\\1 & 0 & 0 & -1\\0 & 1 & 1 & 0\end{bmatrix}$$

注意,矩阵 $\boldsymbol{T}$ 使得输入输出矩阵方程(10.13)对角化,如下所示:

$$\boldsymbol{BT}=\begin{bmatrix}2/m & 0 & 0 & 0\\0 & 2/m & 0 & 0\\0 & 0 & 2h/J_{xx} & 0\\0 & 0 & 0 & 2h/J_{xx}\end{bmatrix}$$

$$T^{-1}C=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&h&0\\0&0&0&h\end{bmatrix}$$

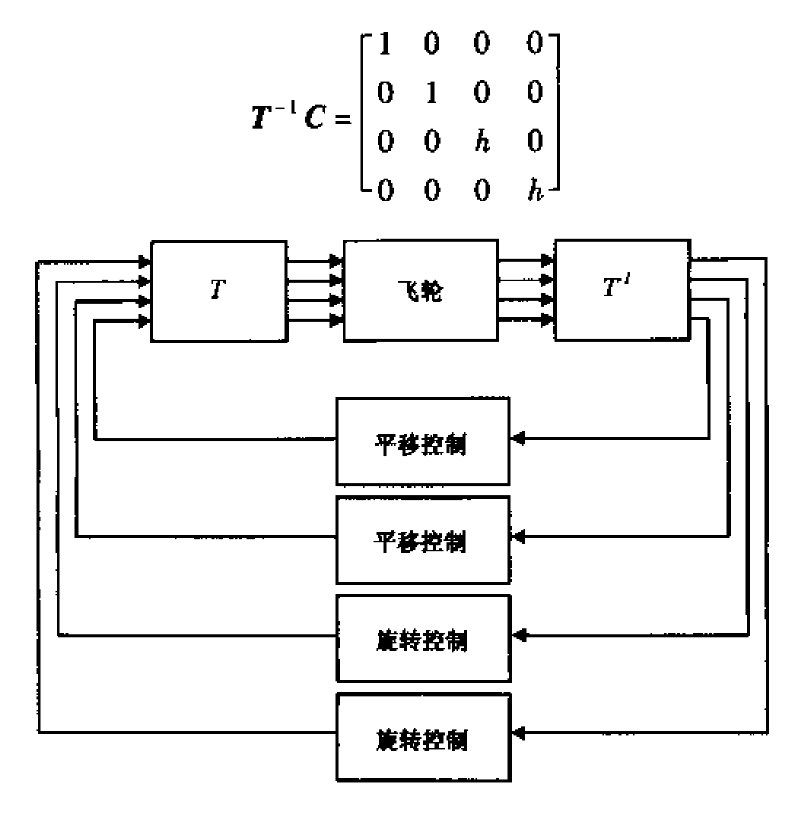


图 10.41 飞轮的静态解耦

用矩阵 $\boldsymbol{T}$ 完成静态解耦之后，即可进行 4 个 SISO 对象的超前设计。图 10.41 说明在实际中如何插入矩阵 $\boldsymbol{T}$。下面的程序可产生解耦后的 SISO 对象，并计算出结果的根轨迹。

```
T=[1 0 0 1;0 1 -1 0;1 0 0 -1;0 1 1 0];
decoupFly=inv(T)*Flywheel(0)*T;
transFly=minreal(decoupFly(1,1));
rotFly=minreal(decoupFly(3,3));
Lead=tf([1 400],[1 1000]);
rlocus(Lead*transFly);
sgrid;
xlabel('Real axis');
ylabel('Imaginary axis');
title('Root locus of translational component');
figure(2);
rlocus(Lead*rotFly);
sgrid;
xlabel('Real axis');
ylabel('Imaginary axis');
title('Root locus of rotational component');
```

程序执行结果如图 10.42 所示。

命令：

```
rlocus(Lead*transFly)
rlocfind(Lead*transFly)
```

用来寻找直线运动系统的稳定增益。

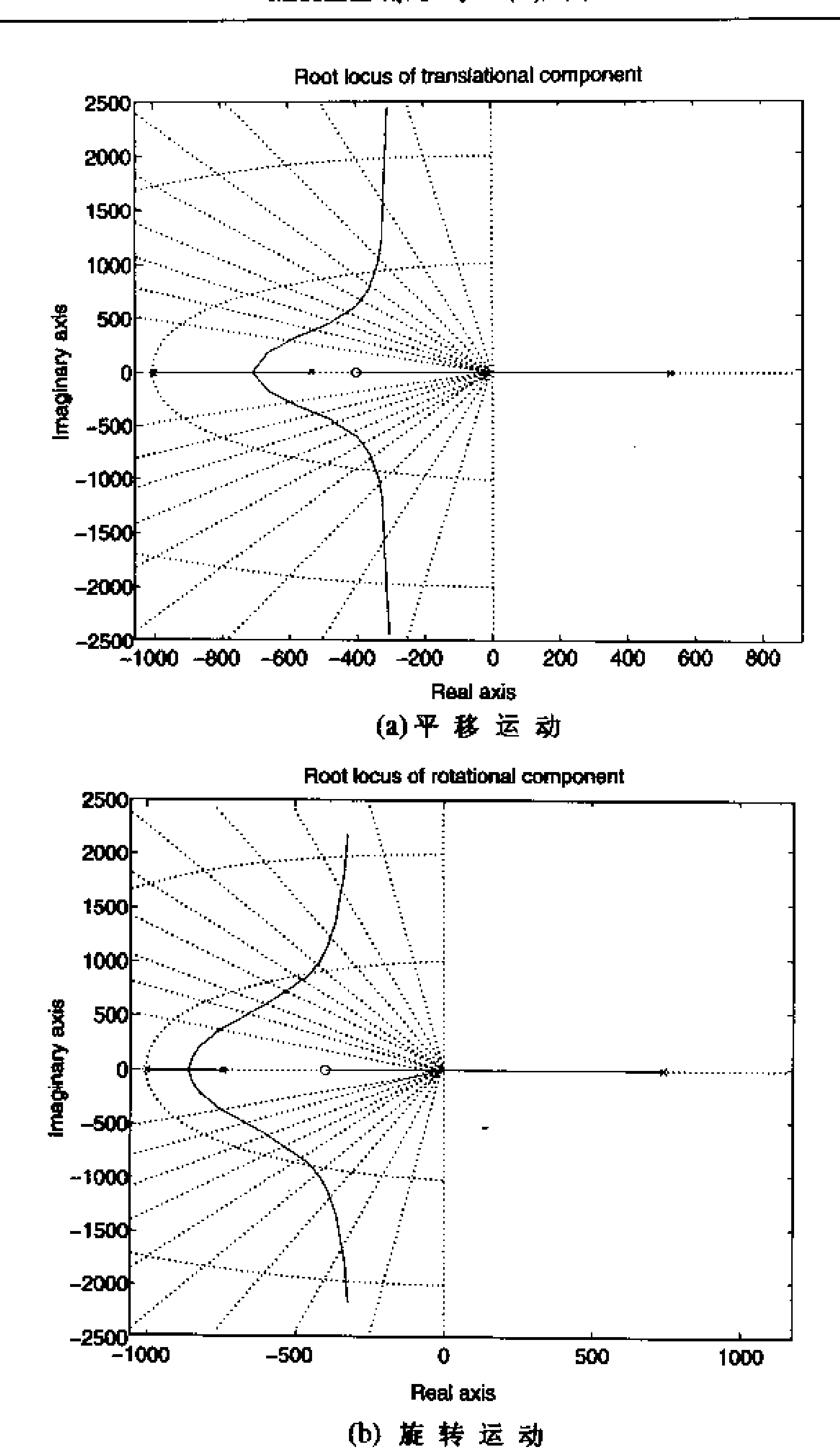


图 10.42 控制回路的根轨迹

用类似的命令：

```
rlocus(Lead * rotFly)
rlocfind(Lead * rotFly)
```

寻找旋转系统的稳定增益。通过设计，用相同增益值(5)可同时满足系统中的平移和旋转两个组件的要求。既然平移和旋转控制的增益是相同的，则控制器也是相同的。由于矩阵 T 已折算到了控制器的传递函数之中，所以实现时不需要再进行解耦。这使得实现过程更加容

易。

现在为飞轮设计 LQG 控制器。设计包含选择适当的代价函数 R。下面的程序在 $10^{-10} \leqslant R \leqslant 10^{-6}$ 时计算最优闭环极点。程序执行结果如图 10.43 所示。

```
[A,B,C,D] = ssdata(Flywheel);
clPoles = [ ]
R = logspace( - 10, - 6,60);
for i = 1:length(R)
 [K,S,E] = lqr(A,B,C' * C,R(i) * eye(4));
 clPoles = [clPoles,E];
end;
plot(real(clPoles),imag(clPoles),'kx');
sgrid;
title('Optimal root locus for the Flywheel system');
ylabel('Imaginary axis');
xlabel('Real axis');
text( - 700,0,'R = 10^{ - 10}');
text( - 1000,1000,'R = 10^{ - 6}');
```

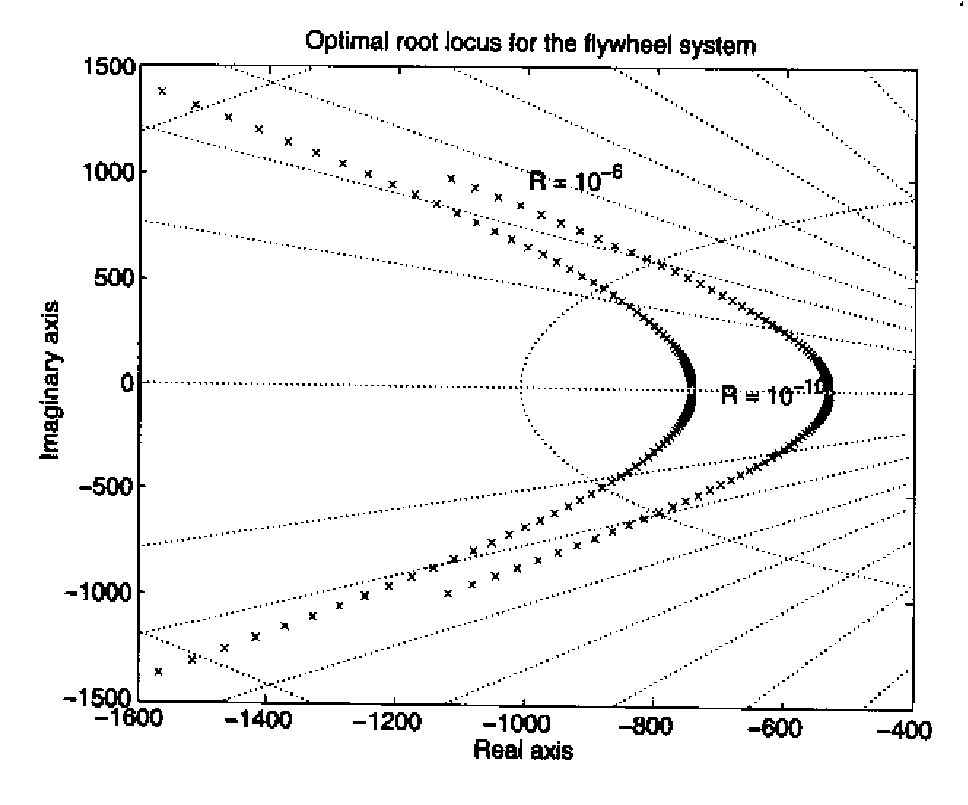


图 10.43 $10^{-10} \leqslant R \leqslant 10^{-6}$ 时飞轮的最优根轨迹

为比较两种不同的控制方法,在相同的初始条件下,下面的程序计算了两个不同控制器的响应。飞轮略微上倾 $\phi(0) = 0.001$ r,其他初始条件均为零。尽管控制器按 0 r/m 设计,但飞轮的转速仍然为 10 000 r/m。基于系数 $R = 10^{-6}$ 设计 LQG 控制器。

```
Control = 4e4 * eye(4) * tf([1 400],[1 1000]);
x0 = zeros(12,1);x0(3) = 1e - 3;t = linspace(0,0.25,1000);
y1 = initial(feedback(Flywheel(10000),Control),x0,t);
[A,B,C,D] = ssdata(Flywheel);
K = lqr(A,B,C' * C,1e - 6 * eye(4));
```

```
L=(lqr(A',C',B*B',1e-6*eye(4)))';
ControlSS=reg(Flywheel,K,L);
x0=zeros(16,1);x0(3)=1e-3;
ys=initial(feedback(Flywheel(10000),ControlSS,+1),x0);
plot(1000*yl(:,1),1000*yl(:,2),'k-',1000*ys(:,1),1000*ys(:,2),'k-');
grid;
xlabel('x upper bearing');
ylabel('y upper bearing');
title('LQG versus lead control of the flywheel')
legend('Lead Control','LQG Control',4)
```

程序执行结果如相位图 10.44 所示。两个控制器都具有相同的控制性能使飞轮保持悬浮。

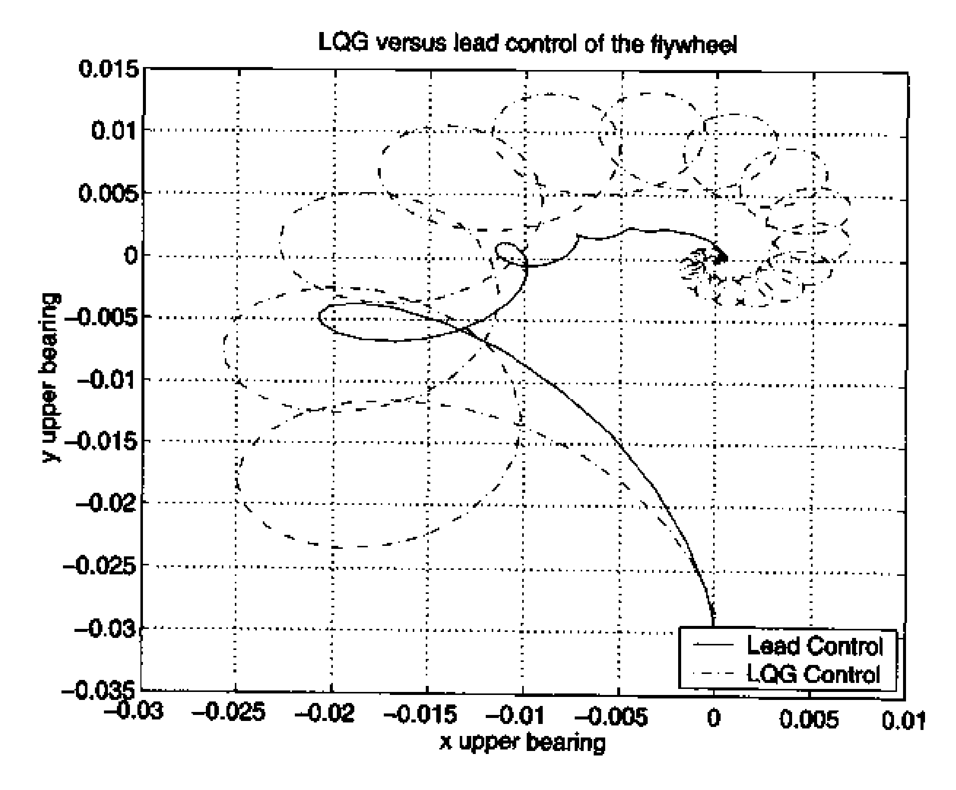


图 10.44 状态空间和超前控制飞轮的相位响应

练习

10.1 悬浮系统① 如图10.45所示，输入 $y(t)$为路面的水平位置，输出 $x(t)$为 m_1 的绝对位置。系统的传递函数为：

$$\frac{y(s)}{r(s)}=\frac{sbk_2+k_1k_2}{m_1m_2s^4+b(m_1+m_2)s^3+k_1(m_1+m_2)s^2+k_2m_1s^2+k_2bs+k_1k_2}$$

假设 $m_1=500$ kg, $m_2=100$ kg, $b=1\,000$ Ns/m, $k_1=2\,000$ N/m, $k_2=10^4$ N/m。

(a) 板式路面可近似为：

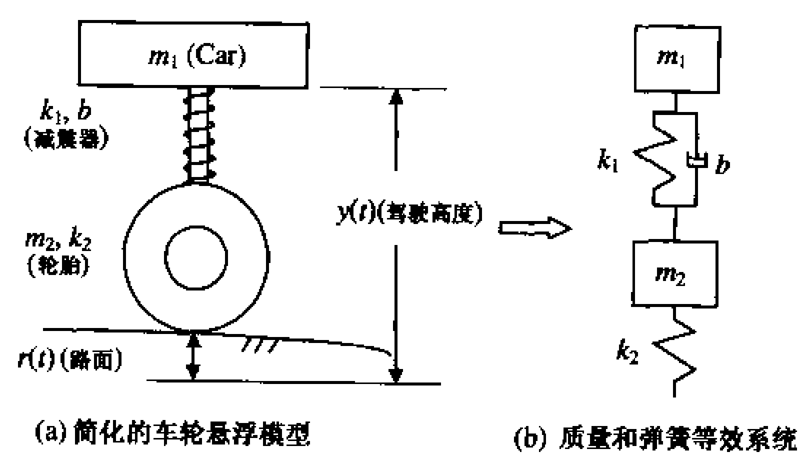
$$r(t)=\varepsilon\sin\omega t$$

① 参见 U. Ozguner, H. Goktas 和 H. Chan, "Automative Suspension Control Through a Computer Communication Network," *Proceedings of 1st IEEE Conference on Control Application*, 第 2 卷, 1992 年, 第 895 - 900 页。

当 $y(t)$振幅是 $r(t)$振幅的 10% 时,确定 ω 值。

(b)在实际中,汽车的质量增加一倍。在 500 kg$\leqslant m_1 \leqslant$1 000 kg 的范围内,将波特图幅值作为 m_1 的函数,生成网格表面图。

(c)ω 作为 b 和 m_1 的函数,当道路变化 $r(t)$减为 90% 时,画出 ω 的值。



(a) 简化的车轮悬浮模型　　(b) 质量和弹簧等效系统

图 10.45　悬 浮 系 统

10.2 构建一个 5 阶系统,其极点等间隔分布在左半平面直径为 $2\pi k$ 的圆上,5 个零点在右半平面的对称位置。令直流增益为 1,且 $k=1$。

(a)将 5 阶系统与一个简单的一阶对象串联,一阶对象极点为 −1,直流增益为 1。在同一张图上,画出时间为 4 s 的系统阶跃响应曲线。

(b)计算超调量和上升时间。

(c)当 $k=0.5$ 及 $k=2.0$ 时,重复(a)和(b)两步。观察随着 k 的增加,响应延迟增加的情况。

(d)当有 3 个和 7 个等分极点时,重复(a)和(b)两步。观察随着极点数目的增加,振荡增大的情况。

10.3 用下面的简单常态模型设计控制器①:

$$G_0(s)=\frac{1}{s+1}$$

对象不能被精确地模型化。对应下面给出的几个模型,画出常态模型 $G_0(s)$的开环阶跃响应和相应对象模型的开环阶跃响应。画出采用比例误差控制,且增益为 20 的闭环阶跃响应。最大时间为 2 s。注意,尽管开环响应差别很大,但闭环响应几乎相同。

(a)
$$G_1(s)=\frac{3.7}{0.75s+0.6}$$

(b)
$$G_2(s)=\frac{1.63}{0.94s+0.92}$$

(c)
$$G_3=\frac{0.7s^2+7s+17}{s^3+2s^2+5.2s+4}$$

① 参见 R. Jurgen, *Electronic Engine Control Technologies*, SAE International, Troy, MI, 1999 年。

10.4 研究第 10.5.1 节讨论的弹性驱动轴系统，如图 10.25 所示。很难准确地知道谐振频率模型，对谐振频率的估计可高可低。为了研究这个问题，将没有补偿的指针波特图与带有以下两种补偿形式的波特图画在一起：

$$带阻滤波器 \# 1: \frac{(s+3+28i)(s+3-28i)}{(s+60)^2}$$

$$带阻滤波器 \# 2: \frac{(s+3+34i)(s+3-34i)}{(s+60)^2}$$

在画图以前令每个带阻滤波器增益为 1。

10.5 实际系统中通常存在非线性。研究图 10.46 所示的非线性输入反馈系统。非线性遵循关系式 $u(t)=f(e(t))$，其中，$e(t)$是输入信号，$u(t)$是非线性输出信号。对象为：

$$G_0(s)=\frac{1}{s+1}$$

图 10.46 非线性输入的简单系统方框图

对于下面的非线性函数，当控制器增益分别为 $k=1$，$k=10$ 及 $k=100$，且输入为阶跃信号时，求系统稳态响应。

(a) $$f(e)=0.2(e^3-e)$$

(b) $$f(e)=e+\sin(e) \quad |e|>1$$
$$=0 \quad |e|\leqslant 1$$

(c) $$f(e)=\arctan(e)$$

10.6 研究图 10.47 所示的导弹，空气的侧向力使导弹相对于它的重心旋转，空气产生的力可视为作用于压力中心的一个点力。如果压力中心在重心的前面，则导弹是不稳定的。系统的输入是推力角度 $\Psi(t)$，系统的输出是 $\theta(t)$。空气的阻力可以建模为 $F_d=k_d\sin(\theta)$，其中 k_d 依赖于火箭的形状和速度。由火箭发动机产生的外力为 $F_r\sin(\Psi)$。其他相关参数是：l_1 为从火箭的发动机到导弹重心的距离，l_2 为从导弹重心到压力中心的距离，J 是火箭的转动惯量。

对于固定的 k_d 和 F_r，当 Ψ 和 θ 很小时，从 Ψ 到 θ 的传递函数为：[①]

$$G(s)=\frac{l_1F_r/J}{s^2-l_2k_d/J}$$

假设 $l_1F_r/J=l_2k_d/J=9$ 为关心的工作点。

(a)用超前控制结构：

$$C(s)=k\frac{s+z}{z+p}$$

求使闭环系统稳定的 k、z 及 p 的值。

① 参见 M. Driels, *Linear Control Systems*, McGraw - Hill, New York, 1996 年。

(b)当火箭的速度变化时，k_d 也变化。在火箭的运行中速度变化是很大的，实际上也改变了 k_d。对(a)中设计的控制器，用根轨迹图确定使系统稳定的 k_d 的范围。

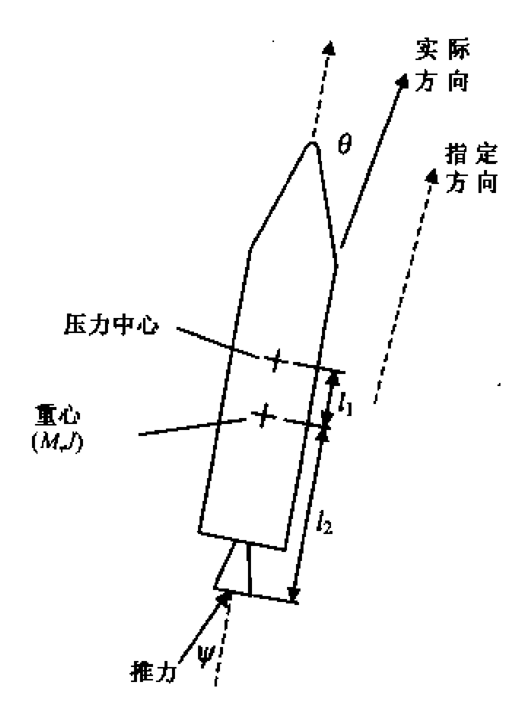


图 10.47　导弹的姿态控制

10.7 设计汽车巡行控制器。汽车可以建模为一个具有限制前行阻尼器的物体。从发动机调速汽门角到汽车速度的开环传递函数 $G(s)$为：

$$G(s)=\frac{1}{ms+b}$$

$C(s)$为 PI 控制器，传递函数为：

$$C(s)=k_p+k_i/s$$

假设汽车的质量 $m=1\,200$ kg，摩擦系数 $b=70$ Ns/m，路的斜度作为扰动输入。系统的方框图如图 10.48 所示。期望的速度为 $r(t)$，电动机的输入(调速汽门角度)为 $u(t)$，用车速表测得的车速为 $y(t)$。扰动 $d(t)$描述了路面斜度的影响程度。

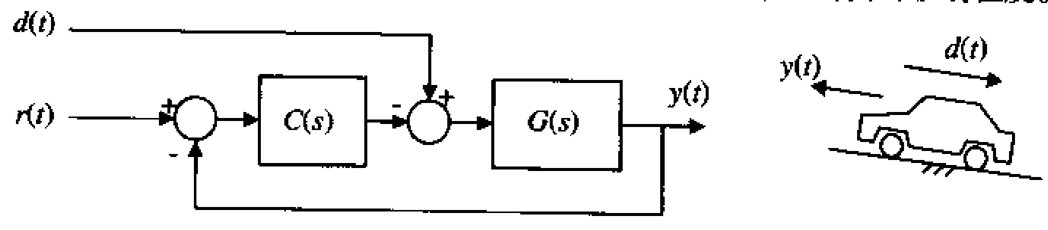


图 10.48　汽车巡行控制方框图

(a)从扰动 $d(t)$到速度 $y(t)$ 的传递函数为：

$$G_{cl}(s)=\frac{G(s)}{1+C(s)G(s)}$$

其中 $G(s)$为开环响应，$C(s)$为前面定义的 PI 传递函数。在同一图上画出 $k_p=100$ 及 $k_i=0$ 时系统开环及闭环的阶跃响应曲线。注意随着增益的增大扰动

在减小。

(b)从速度目标值 $r(t)$到速度输出 $y(t)$的传递函数为：

$$G_{ry}(s)=\frac{C(s)G(s)}{1+C(s)G(s)}$$

画出 $k_p=100$ 及 $k_i=0$ 的阶跃响应曲线。

(c)比例增益 P 的变化范围在 50 ~ 150 之间，画出(a)和(b)阶跃响应的系统稳态输出。可以看出积分器的作用消除了稳态误差，并且积分器的速度由增益 k_i 确定。

10.8 一般的汽车减震系统由弹簧和阻尼器构成，是一个被动系统。为了提高汽车的减震性能必须采用主动的减震系统。图 10.49 显示了一个汽车车轮的主动减震系统的简化模型，其中 $x(t)$代表路面输入，$y(t)$是车厢的垂直位置。假设轮胎的质量可以忽略，采用速度反馈，$u(t)=C\mathrm{d}x(t)/\mathrm{d}t$。调节器作用在支撑杆与车厢上的力与速度成比例：$C(\mathrm{d}x/\mathrm{d}t-\mathrm{d}y/\mathrm{d}t)$。从 $y(t)$到 $r(t)$的传递函数为：

$$G(s)=\frac{(c+b)s+k}{ms^2+(c+b)s+k}$$

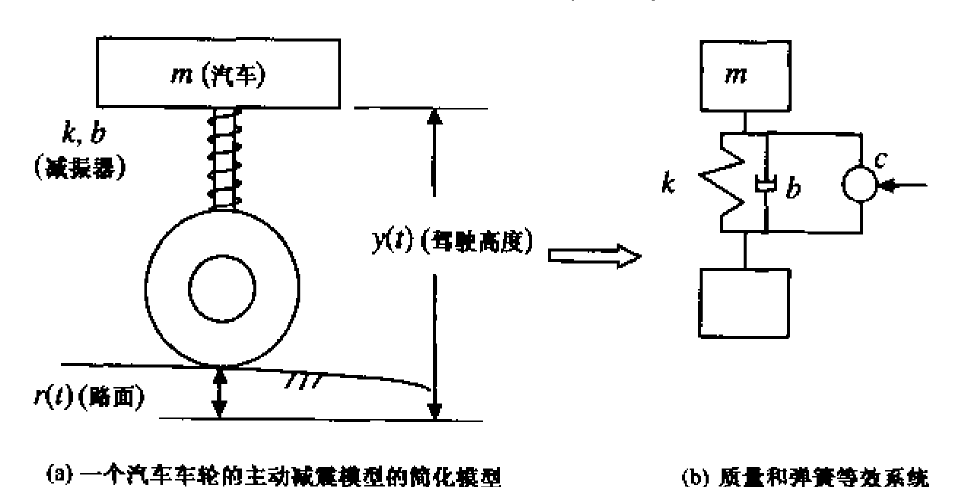


图 10.49　汽车减震系统示意图

(a) 用函数 rlocus 和 rlocfind 确定 c 的值，使阻尼系数为 1，即系统处于临界阻尼状态。用函数 rlocus 时注意，分母可以写成 $ms^2+bs+k+cs$。rlocus 的标准形式为：分子[1 0]和分母[m b k]。由 rlocfind 确定的值是 c。把十字线放在闭环极点与实轴首次相交的位置。

(b) 分别画出系统带有主动控制及没有主动控制($c=0$)的阶跃响应和波特图。假设 $m=5\,000$ kg，$k=8\times10^5$ N/m，$b=12\,000$ Ns/m。

(c) 在这个问题中弹簧用 k 计算。采用(b)中的设计，k 的范围在 4×10^5 到 10×10^5 N/m 之间，生成阶跃响应的网格图形。比较两个系统的性能。

10.9 为大型慢速行驶的船设计自动驾驶仪，用罗盘的输出作为反馈，控制器发出命令给方向舵机构，命令经过延迟变成期望的位置，从而调整船的运行。对船的 Nomoto 方

程[1] 线性化得出下面的方程。没有控制器时操纵系统的开环传递函数为[2]：

$$G(s)=\frac{s+0.03}{s(s+0.09)(s+0.04)(s-0.0004)}$$

(a) 对象有不稳定极点，画出操纵系统的根轨迹。

(b) 用超前控制结构：

$$C(s)=k\frac{s+z}{z+p}$$

要求闭环系统响应稳定，超调小于 30%。求 k、z 和 p 的值。

(c) 若用传感器提供速度信息，则可以使用 PD 控制。因此，$C(s)=k_p+sk_d$，当闭环响应稳定，超调小于 5%，调节时间小于 275 s 时，求 kp 和 kd 的值。

10.10 一种带催化转换器的新汽车模型能使废气排放量达标。催化转换器要求严格控制发动机的空气/燃料比(A/F)、点火时间和废气再循环。研究空气/燃料比调节任务，以 A/F 作为输出，炭化器的传递函数如下[3]：

$$G(s)=\frac{4e^{-T_d^s}}{s+4}$$

其中，时间延迟 T_d 为 0.2 s，函数 pade 求取 T_d 的近似值，也可以将传递函数的输出延迟设为 0.2，因为函数 pade 是 MATLAB 中专门用于时间延迟计算的，所以使用起来很方便。

(a) 假设控制器设计中忽略时间延迟，令控制器为 PI 控制，可写为：

$$C(s)=k_p+\frac{k_i}{s}$$

令 $k_i=2$ 并选择 k_p 的值，使得单位阶跃响应的上升时间小于 0.4 s。确定系统的阶跃响应。

(b) 研究将时间延迟信号馈送到控制器的过程，如图 10.50 所示。控制器中的附加补偿单元包括对象模型及其时延，被称为史密斯(*Smith*)预估器。采用超前控制机构。

$$C(s)=k\frac{s+z}{z+p}$$

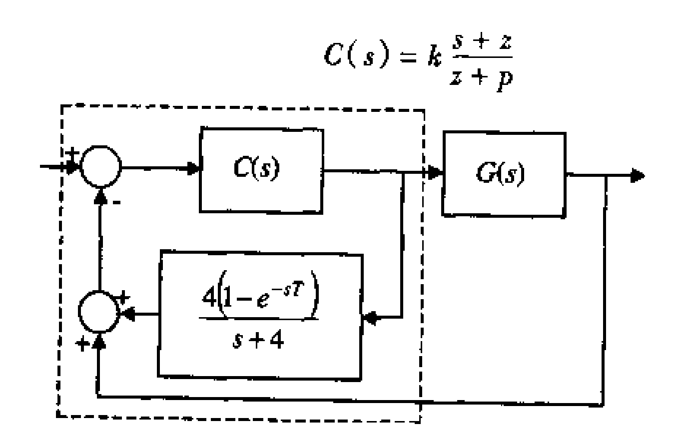


图 10.50 带有史密斯预估器的废气控制系统

---

① 参见 M Dnels，出处同上。

② 参见 C L Phillips 和 R D. Harbor, *Feedback Control Systems*，第三版，Prentice Hall, Englewood Cliffs, NJ, 1996 年。

③ 参见 B. Kuo, *Automatic Control Systems*，第七版，Prentice Hall, Englewood Cliffs, NJ, 1995 年，第 815 页

要求闭环系统稳定且没有超调,上升时间小于 0.2 s,求 k、z 和 p 的值。将结果与(a)中的 PI 控制器进行比较。

(c) 假设时间延迟和对象都没有精确模型,确定系统延迟时间为 0.3 s 和 0.1 s 的阶跃响应。控制器采用(b)中所设计的控制器,对象模型的直流增益为 1.2,极点不是 -4 而是 -5。从结果看,对延迟时间估计是高点好还是低点好?

10.11 研究图 10.51 所示电动自行车方框图。倾斜仪是用来检测车体偏离垂直方向角度 $y(t)$ 的。倾斜仪的输出与期望垂直角度 $r(t)$ 相比较,将误差送回控制器并产生操纵信号。系统的任何扰动 $d(t)$ 都作为车子的输入。方框图 10.51 的传递函数为:

$$G(s) = \frac{9}{s^2 + 9}$$

$$F(s) = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}$$

$$C(s) = k\frac{s + z}{s + p}$$

其中 $C(s)$ 为超前控制器。

(a) 微型机械倾斜仪的调节时间为 0.2 s,带宽为 125 Hz。传感器的参数 $\omega = 250\pi$,$\zeta = 20/\omega$。当单位阶跃响应的超调小于 20%,调节时间小于 4 s 时,求 k、z 和 p 的值。

(b) 假设有另一种类型的倾斜仪可以选择,它基于钟摆的原理,它的谐振频率是 7.4 Hz($\omega = 14.8\pi$),阻尼系数 $\zeta = 0.4$。这样的倾斜仪可以用吗?

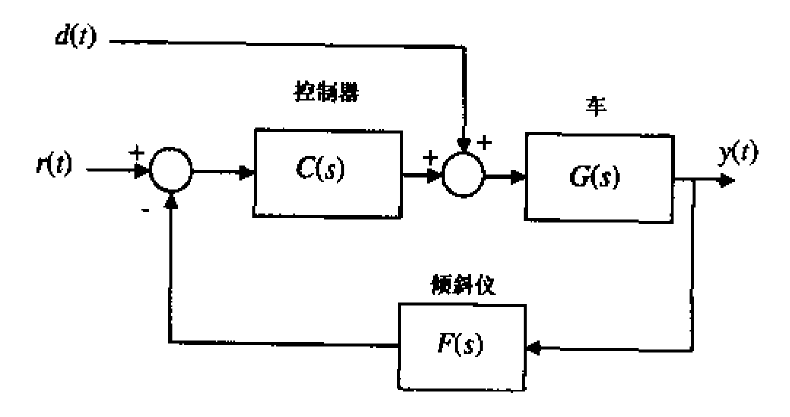


图 10.51 电动自行车的方框图

10.7 参考文献

1. Anand D K and Zmood R B. Introduction to Control Systems, Oxford: Butterworth and Heinmann, Ltd., 1995.
2. Chowanietz E. Automobile Electronics. SAE International. MI: Troy, 1995.
3. Dorf R and Bishop R. Modern Control Systems. MA: Addison-Wesley Publishing, Reading, 1997.
4. Driels M. Linear Control System Engineering. New York: McGraw-Hill, 1996.
5. Franklin G, J Powell and Emami-Naeini A. Feedback Control of Dynamic Systems. 3rd ed. MA:

Addison-Wesley, Reading, 1994.

6. Friedland B. Advanced Control System Design; NJ: Prentice Hall, Englewood Cliffs, 1996.

7. Jurgen R. Electronic Engine Control Technologies. MI: SAE International. Troy, 1999.

8. Kuo B. Automatic Control Systems. NJ; Prentice Hall, Englewood Cliffs, 1995.

9. Levine W. The Control Handbook. FL: CRC P, Boca Raton, 1996.

10. Nise N. Control Systems Engineering. MA: Addison-Wesley, Reading, 1995.

11. Ozguner U, H Goktas and Chan H. Automotive Suspension Control Through a Computer Communication Network. Proceedings of 1st IEEE Conference on Control Application, vol. 2, 1992: 895—900.

12. Phillips C and Harbor R. Feedback Control Systems. NJ; Prentice Hall, Englewood Cliffs, 1996.

第11章 流 体 力 学

本章对流体力学中的几类问题加以分析,并用不同可视化技术表示几种流场。

11.1 流体静力学

在流体静力学中,当高度一定时,任意平面上均质液体的静压强是恒定不变的。高度一旦变化则压强也随之而变化。满足以下方程:

$$\frac{\mathrm{d}P}{\mathrm{d}z} = -\rho g \tag{11.1}$$

式中,$\rho=\rho(z)$是密度分布函数,g 是重力加速度(9.81 m/s^2),P 是压强,z 是笛卡儿坐标的垂直分量,向上为正。利用式(11.1)解决两个流体静力学问题。

11.1.1 标准大气压下的压强分布

在标准大气压下温度分布呈分段线性,这种线性分布可由不同海拔高度下的一组温度值定义,表 11.1 体现了这一规律。假定海拔高度与温度呈线性变化,且空气为理想气体,则为:

$$\rho = P/(RT)$$

式中 $R=287.13$ J/(kg K),是理想气体常数,T 是热力学温度,单位为开尔文。在这种条件下,结合式(11.1),可以得到:

$$P(z) = P_0\exp\left[-\frac{g}{R}\int_0^z\frac{\mathrm{d}z}{T}\right] \tag{11.2}$$

式中在 $z=0$ 处的压强 $P_0=101\ 330$ Pa。

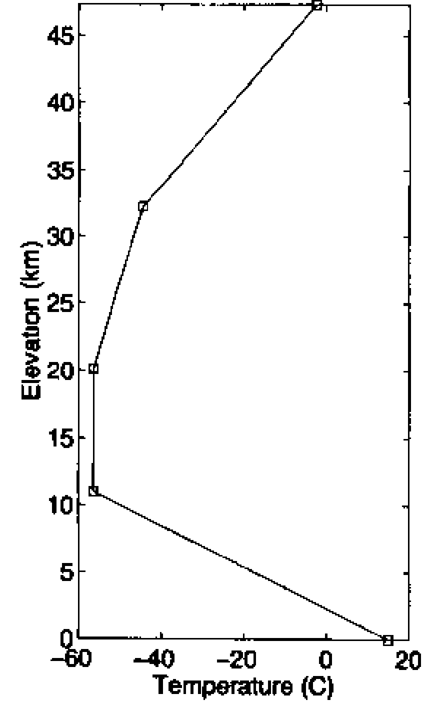
表 11.1 标准大气压下的温度与海拔高度的依变关系

海拔高度(m)	温度(℃)
0.0	15 0
11 000.0	-56.5
20 100.0	-56.5
32 200 0	-44 5
47 300.0	-2 5

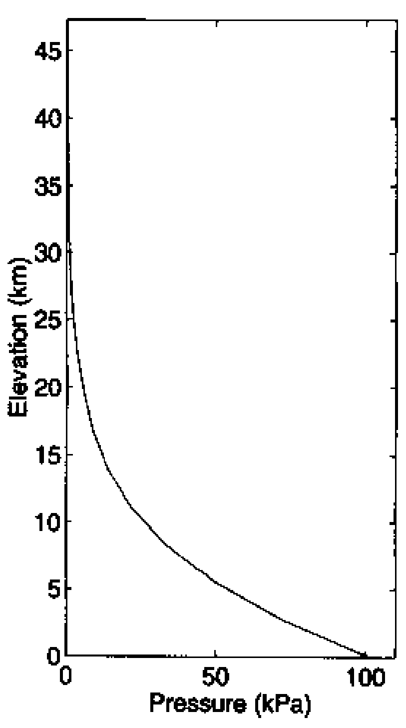
虽然式(11.2)很容易求解,但是必须对数值进行估测。下述程序对式(11.2)求解,并画出 P和 T随 z变化的函数曲线,用函数 interpl 实现表 11.1 中温度数值的线性插值。结果如图 11.1 所示。

```
gravity = 9.81; p0 = 101330; R = 287.13;
tempC = [15 - 56.5 - 56.5 - 44.5 - 2.5];
z = [0 11000 20100 32200 47300];
inversetemp = 1./(tempC + 273.15);
np = 18;
```

```
goverR = gravity/R;
elevation = linspace(0,z(end),np);
pressure = zeros(1,np);
intarg = inline('interp1(z,inversetemp,elevation)','elevation','z','inversetemp');
for i = 1:np
   pressure(i) = p0 * exp( - goverR * quad8(intarg,0,elevation(i),[],[],z,inversetemp));
end
subplot(1,2,1)
plot(tempC,z/1000.0,'k-s')
axis([ -60 20 0 elevation(end)/1000.0])
xlabel('Temperature(C)')
ylabel('Elevation (km)')
subplot(1,2,2)
plot(pressure/1000.0,elevation/1000.0,'k')
axis([0 110 0 elevation(end)/1000.0])
ylabel('Elevation (km)')
xlabel('Pressure(kPa)')
```



(a) 表11.1中温度与海拔高度的依变关系

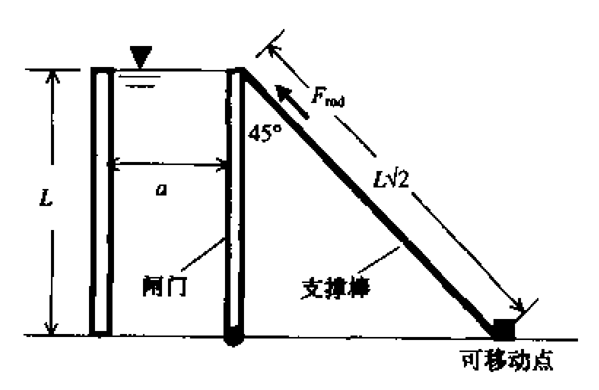


(b) 式(11.2)中的压强与海拔高度的依变关系

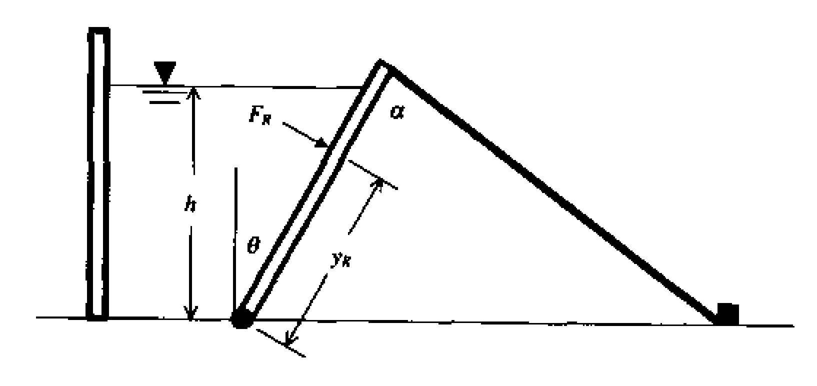
图11.1 标准大气压

11.1.2 平面闸门的作用力

研究图11.2所示的水库问题。水库的一个壁是一个可倾斜的金属闸门，它可以绕底部转动，其重量为 W，长度为 L。水库在垂直于书页表面方向上的宽度为 B。初始时闸门垂直放置，水位刚好达到闸门的顶部，水的体积为 $V_w = aLB$。闸门由一根棒支撑，沿棒方向作用在闸门上的力为 F_{rod}，棒的另一端被一障碍物支撑住，这个障碍物可以向右移动，使得闸门能绕转动轴顺时针方向旋转。由于闸门的倾角 θ 只能小于等于某个临界角度 θ_{max}，所以水位只能低于或等于闸门的高度。当 $\theta > \theta_{max}$ 时，水会溢出闸门顶部。



(a)竖 直 的门



(b)开发一定角度 θ 的门

图 11.2 具有可倾斜门的水库

由水库底部、固定的壁、闸门和闸门顶部的平面围成的体积为:

$$\frac{V}{V_w} = \cos\theta + \frac{L}{2a}\cos\theta\sin\theta \tag{11.3}$$

当满足以下条件时,水将溢出水坝:

$$\frac{V}{V_w} < 1.0$$

根据任意角度 θ 下的水的体积与 V_w 的等式关系可以得到水位高度 h 对 θ 的关系式,即:

$$\frac{V_w}{B} = aL = ah + 0.5h^2\tan\theta \tag{11.4}$$

通过对转动轴力矩的测量可以求得 F_{rod} 的值,即:

$$F_{rod} = \frac{F_R y_R + 0.5WL\sin\theta}{L\sin\alpha} \tag{11.5}$$

其中 F_R 是水对闸门的总作用力,y_R 是转动轴到压力中心的距离,角度 α 如图 11.2(b)所示,且由以下方程式给出:

$$\alpha = \theta + \arccos(\cos\theta/\sqrt{2})$$

从流体静力学的平衡关系式中可以得出:

$$F_R = \frac{B\rho gh^2}{2\cos\theta}$$

$$y_R = \frac{h}{2\cos\theta} - \frac{2I_{xx}\cos^2\theta}{Bh^2}$$

式中，

$$I_{xx} = \frac{Bh^3}{12\cos^3\theta}$$

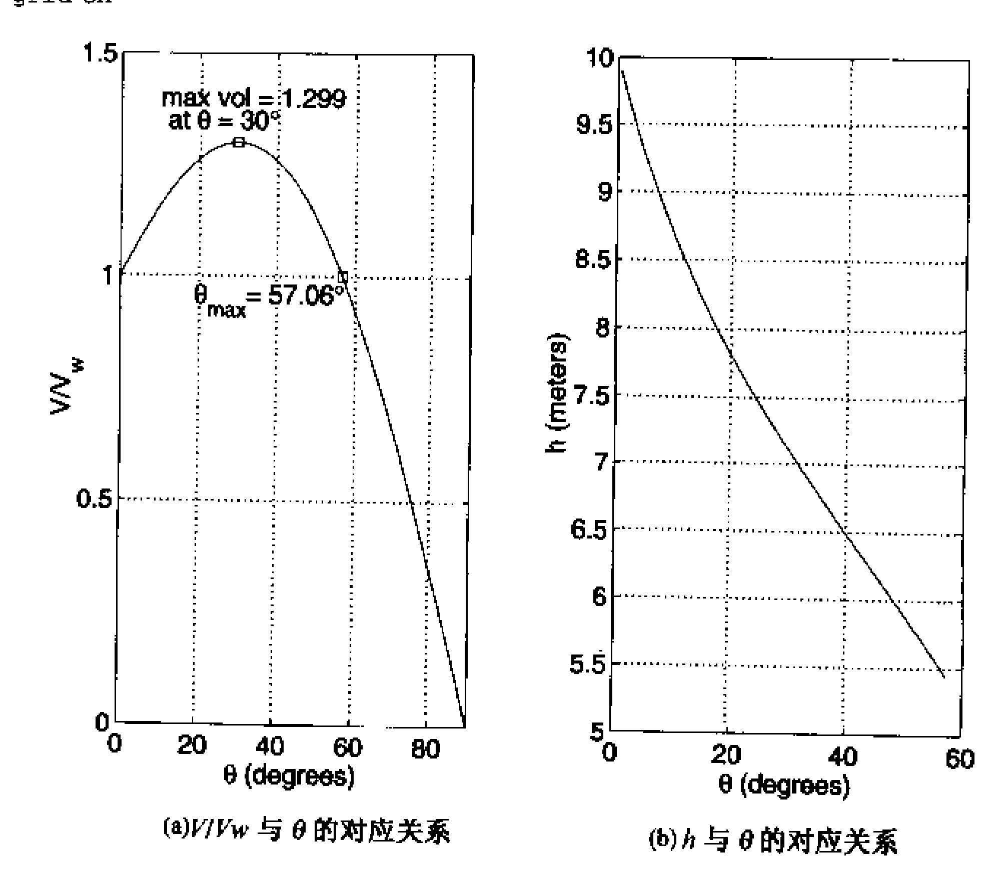
是闸门浸在水中轴的惯性矩。则式(11.5)变为：

$$F_{rod} = \frac{1}{L\sin\alpha}\left[\frac{B\rho gh^3}{6\cos^2\theta} + \frac{WL}{2}\sin\theta\right] \tag{11.6}$$

在以下的程序代码中将求出 θ_{max} 值，并在 $0 \leqslant \theta \leqslant \theta_{max}$ 范围内绘出水深 h 和 F_{rod} 的曲线，结果如图 11.3 所示。假定 $L = 10$ m, $a = 5$ m, $B = 10$ m, $W = 100\,000$ N，此外，V/V_w 由式(11.3)计算得出，可以绘出它对 θ 的关系曲线。MATLAB 中的 fzero 函数可以用来求 θ_{max}，θ_{max} 位置标在图 11.3(a)上。在计算 θ_{max} 的过程中 inline 函数创建 MaxTheta，它的返回值为 $1 - V/V_w$。为了求得 h 对 θ 的关系曲线，用 roots 求解二次方程(11.4)中 h 的正根。水库盛水最多时确定 θ 的值也很有意义，可以由 fminbnd 函数调用 MaxTheta 来实现。求解这些值并显示这些结果(包括图 11.4 中 F_{rod} 对 θ 的关系曲线)的程序代码为：

```
a = 5.0;L = 10.0;B = 10.0;
rho = 1000.0;grav = 9.81;W = 100000.0;
ratio = L/a;
theta = linspace(0.0,pi/2.0);
VoverVw = cos(theta) + 0.5 * ratio * cos(theta). * sin(theta);
options = optimset('display','off');
figure(1)
subplot(1,2,1)
plot(theta * 180/pi,VoverVw,'k')
axis([0.0,90.0,0.0,1.5])
ylabel('V/V _ w')
xlabel('/theta(degrees)')
grid on
hold on
MaxTheta =
inline('1 - (cos(theta) + 0.5 * ratio * cos(theta). * sin(theta))','theta','ratio');
ThetaMaxDeg = fzero(MaxTheta,[0.01,pi/2.0],options,ratio) * 180/pi;
plot(ThetaMaxDge,1.0,'sk')
text(19,0.95,[' \ theta _{max} = ',num2str(ThetaMaxDeg,4)' \ 'circ'])
ThetaMax Vol = fminbnd(MaxTheta,0.0,ThetaMaxDeg * pi/180,options,ratio);
MaxVol = 1 - MaxTheta(ThetaMaxVol,ratio);
plot(ThetaMaxVol * 180/pi,MaxVol,'ks')
text(10,MaxVol + 0.1,['max vol = 'num2str(MaxVol,4)])
text(10,MaxVol + 0.05,['at \ theta = 'num2str(ThetaMaxVol * 180/pi,4)' \ circ'])
subplot(1,2,2)
theta = linspace(0.01,ThetamaxDeg * pi/180);
h = zeros(1,length(theta));
for i = 1:length(theta)
  r = roots([tan(theta(i)) * 0.5,a, - a * L]);
  h(i) = r(2);
end
plot(theta * 180.0/pi,h)
ylabel('h(meters)')
xlabel(' \ theta(degrees)')
grid on
figure(2)
```

```
Frod = ((B * rho * grav * h.^3)./(6 * cos(theta).^2) + 0.5 * W * L * sin(theta))./...
      (L * sin(theta + acos(cos(theta)/sqrt(2))));
plot(theta * 180.0/pi,Frod * 1e - 6)
ylabel('F_{rod}(MN)')
xlabel('\theta(degrees)')
grid on
```



(a)V/Vw 与 θ 的对应关系　　(b)h 与 θ 的对应关系

图 11.3　可转动门构造的结果图

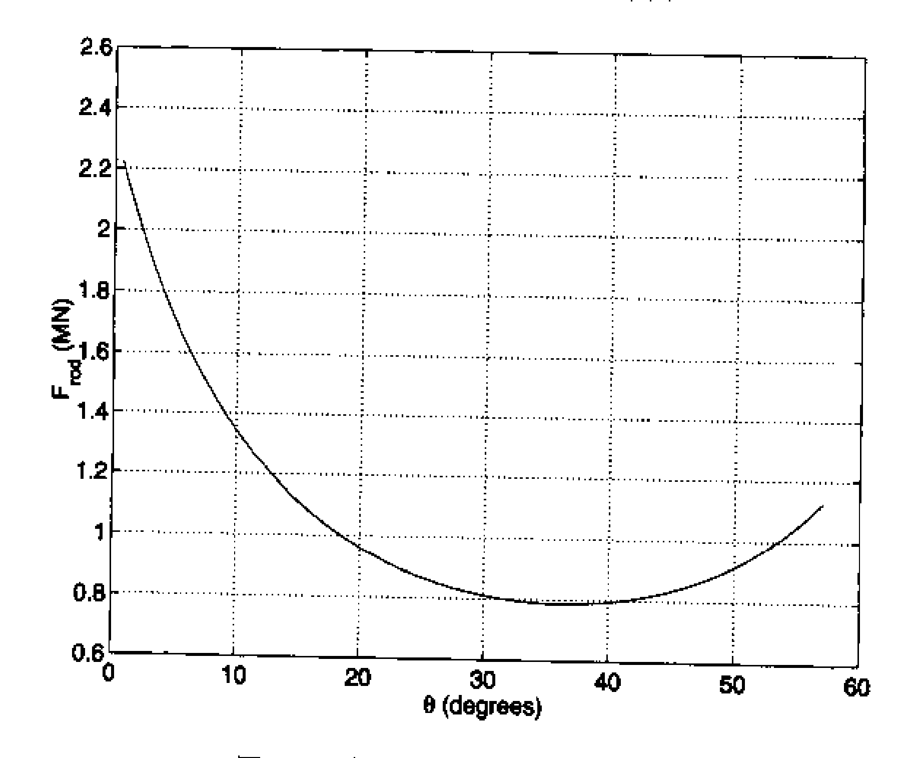


图 11.4　使门关闭的力与 θ 的对应关系

11.2 内部粘性流动

在管道中的层流粘性流动和紊流粘性流动是一大类问题。以下给出了几种问题的解决方法。管道中充分发展低雷诺数的流动为层流，PDE 工具箱可以计算流场及沿管道而降低的压强；高雷诺数的流动为紊流，流动和压强的势差都可以用科尔布鲁克方程式求出。[①]

11.2.1 不同截面水平管道的层流

利用 MATLAB PDE 工具箱，可以较直接地计算出各种形状横截面的水平管道中充分发展的层流场。在大部分教科书中对圆管的分析使用了这种方法。[②] 因此，首先求解这个流场，再将计算出的流速最大值和平均值与解析法求解结果相比较。假定管道半径 R 为 5.0 mm，流体的动力粘性系数 μ 为 0.38 N s/m^2，轴向压强梯度 $\mathrm{d}P/\mathrm{d}x$ 为 1.0×10^8 Pa/m。

轴向流场的微分方程为：

$$\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}=\frac{1}{\mu}\frac{\mathrm{d}P}{\mathrm{d}x} \tag{11.7}$$

式中，$u(y,z)$是 x 轴方向上的速度，y 和 z 是管道横截面的笛卡儿坐标，点(0,0)是管道的中心点，管道壁的坐标(y,z)满足

$$y^2+z^2=R^2$$

壁面处滞流的边界条件为 $u=0$。

下面用 pdetool 来求解这个问题[③]。在 pdetool 中首先创建一个以点(0,0)为中心，以 1.0 为半径的圆，这就意味着创建了一无量纲坐标系 $y'=y/R, z'=z/R$。式(11.7)变为：

$$\frac{\partial^2 u}{\partial y'^2}+\frac{\partial^2 u}{\partial z'^2}=\frac{R^2}{\mu}\frac{\mathrm{d}P}{\mathrm{d}x}=f$$

在 pdetool 中，首先进入 Boundary 菜单选择 Boundary Mode，在圆的四条弧线上确定第立克特边界条件($u=0$)，然后选定椭圆偏微分方程，参数为 $c=1, a=0, f=6\,578.95$ m/s($=(0.005^2/0.38)\times10^8$)。其次，初始化网格并进行两次改进，结果以等高线图形式绘出，如图 11.5 所示。

为了从求解过程中得到其他的定量数据，把求解结果 u 和网格描述的量 p, e, t 一同输出到 MATLAB 命令窗口上，速度的最大值出现在管道横截面的中心点处。为求得该值，可在 MATLAB 命令窗口中输入语句：

```
umax = max(u)
```

结果为 $umax=1\,643.5$，非常接近理论数值 1 644.74。

计算流量 Q 的公式为：

$$Q=\iint_A u\mathrm{d}A\approx\sum_{i=1}^{N_t}u_i\Delta A_i$$

---

① 参见 C.E.Colebrook, Turbulent flow in pipes with particular reference to the transition region between smooth and rough pipe laws, *Journal of the Institute of Civil Engineers*, London, 第 11 卷, 1939 年, 第 133 - 156 页。

② 参见 B.R.Munson, D.F.Young 和 T.H.Okiishi, *Fundamental of Fluid Mechanics*, John Wiley & Sons, New York, 1998 年。

③ 有关如何使用 pdetool 的详细内容参看第 8.7 节。

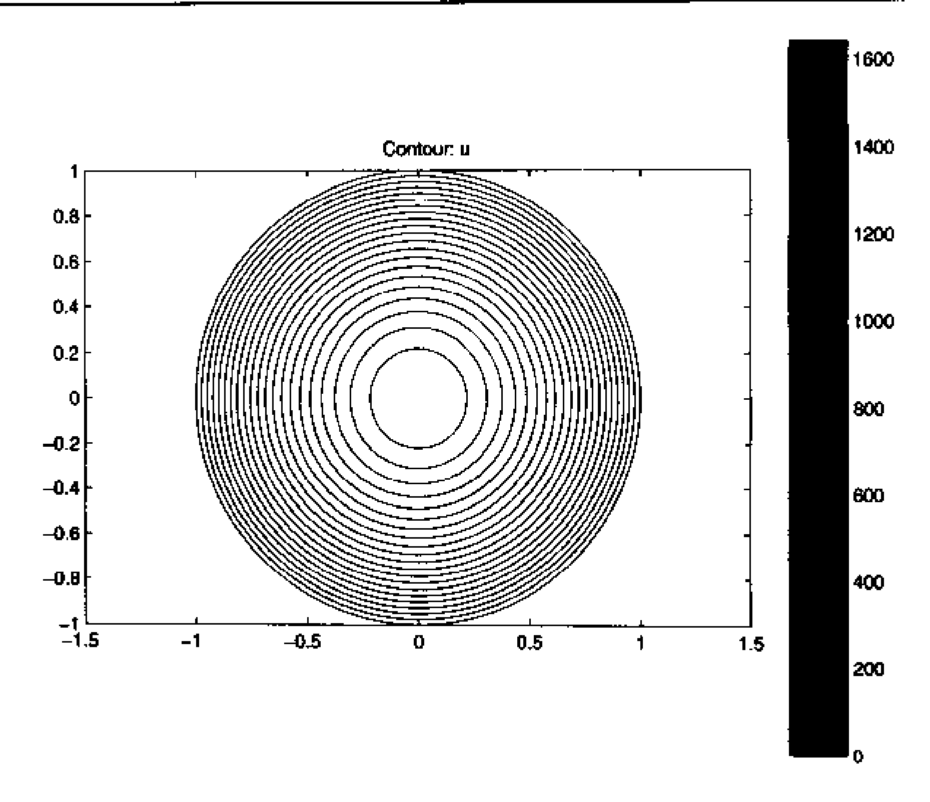


图 11.5 圆管中的轴向流场，最内层等高线的数值为 1 600

式中，A 是管道的横截面积，N_t 是网格中三角形的数量，u_t 为每个格状三角形的中心处的速度，ΔA_t 是每个格状三角形的面积。在以下的程序代码中，用 pdeintrp 函数确定 u_t，用 pdetrg 函数确定 ΔA_t。

```
ui = pdeintrp(p,t,u);
DeltaAi = pdetrg(p,t);
Q = sum(DeltaAi. * ui)
```

把 u、p、e 和 t 输出到 MATLAB 窗口后，执行上述程序得到 $Q = 2\,580.6$，平均速度 $u_{ave} = Q/A = 2\,580.6/\pi = 821.4$，非常接近最大速度理论值的一半，即 1 644.74/2 = 822.37。对网格做进一步改进就会使计算值和理论值更加一致。

也可以绘出速度与管道直径的函数图形：$y' = 0, -1 \leqslant z' \leqslant 1$。为此，首先必须用 tri2grid 函数把三角网格数据添加到位于管道横截面半径的点上，结果如图 11.6 所示。

图中曲线具有理论方法得到的抛物线型轮廓。其实现代码为：

```
z = linspace( - 1,1,25);
uyz = tri2grid(p,t,u,0,z);
plot(uyz,z)
axis([0,1800, - 1.0,1.0])
ylabel('z')
xlabel('Axial velocity')
```

11.2.2 垂直管道中向下的流动

一个立于垂直方向上的光滑管道，长度为 L，直径 $D = 4.0$ cm。管道中通过密度为 $\rho = 1\,000.0$ Kg/m^3，运动粘度 $v = 1.2 \times 10^{-6}$ m^2/s 向下流的水（如图 11.7 所示），求其流动速率。对于流动速率来说，重力引起的压强升高与流动引起的压强降低相互抵消，即管道中的静态压强与

管道长度无关。

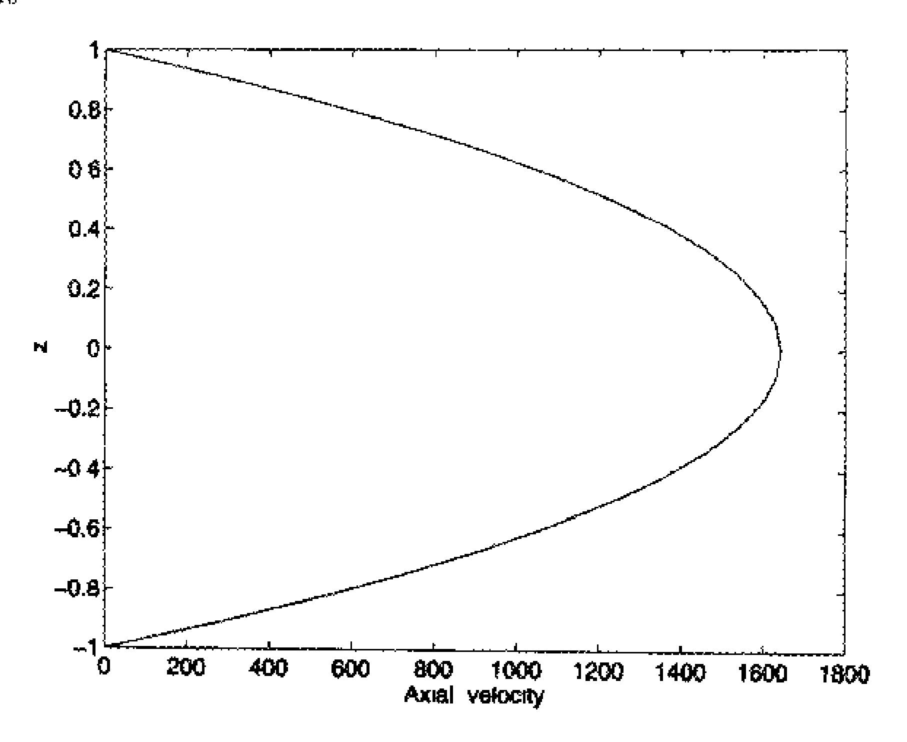


图11.6 管道内的轴向流速与半径位置的关系

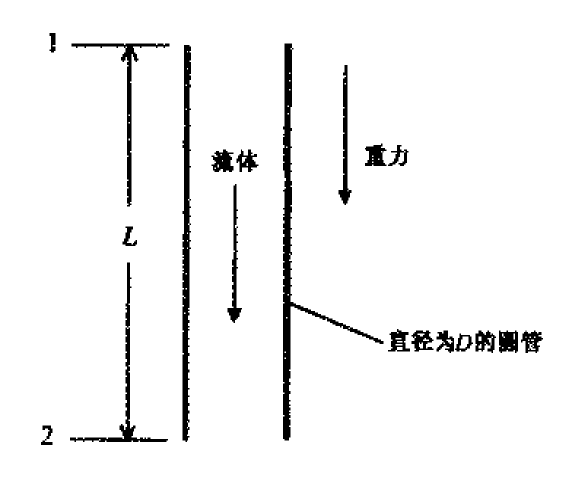


图11.7 直径为 D 的垂直管道内的流体

水头损失公式为：

$$\frac{P_1}{\rho g}+\frac{V_1^2}{2g}+z_1=\frac{P_2}{\rho g}+\frac{V_2^2}{2g}+z_2+\frac{\lambda L V^2}{2gD} \tag{11.8}$$

式中 P 是压强，V 是平均流动速度，z 是高度，λ 是摩擦系数。该问题中 $P_1=P_2$，$V_1=V_2$，$z_1-z_2=L$；因此，式(11.8)可简化为：

$$\lambda=\frac{2gD^3}{v^2R_e^2} \tag{11.9}$$

式中，R_e 是雷诺数，可定义为：

$$R_e=\frac{VD}{v} \tag{11.10}$$

不同粗糙度(k/D)管道的 λ 与 R_e 的阔尔布鲁克关系式为(参见练习5.5)：

$$\frac{1}{\sqrt{\lambda}} = -2\log_{10}\left(\frac{2.51}{R_e\sqrt{\lambda}} + \frac{k/D}{3.7}\right) \quad R_e \geqslant 4\,000 \tag{11.11}$$

该例中，$k=0$。消去式(11.9)和式(11.10)中的 λ 可以得到管流的雷诺数和期望流动速率的一个经验公式。由 ColebrookFriction 函数估算科尔布鲁克方程表达式，函数如下：

```
function value = ColebrookFriction(Re,nu,kOverD,gravity,diameter)
lambda = 2 * gravity * diameter^3/(nu * Re)^2;
value = 1/sqrt(lambda) + 2 * log10(kOverD/3.7 + 2.51/(Re * sqrt(lambda)));
```

程序代码为：

```
diameter = 0.04;gravity = 9.81;
nu = 1.2e - 6;kOverD = 0.000;
options = optimset('display','off');
Re = fzero('ColebrookFriction',[1e3,1e7]),options,nu,kOverD,gravity,diameter;
disp(['Re = ',num2str(Re)])
disp(['Flow Rate = 'num2str(pi * diameter * Re * nu/4)'m^3/s'])
```

执行结果为：

```
Re = 240405.8408
Flow Rate = 0.0090631 m^3/s
```

流量为 $Q=\pi D^2 V/4$。

11.2.3 三水库问题

三水库问题[①]是指三个不同海拔高度的水库连接到同一个交叉点 J 处(如图 11.8 所示)。若给定管道的下列参数：管道的长度 L_j，直径 d_j，粗糙度 k_j 和各个水库的高度 h_j，则可确定每个管道的相应流速 Q_j 和流向。方法如下：如果在交叉点 J 处安装一个开口的管子，则管子中的水可以升到某个未知高度 h_p，P 点和 J 点的高度差就是交叉点处的压差。其次，在 J 处每个管道的总体流量必为 0，即：

$$\sum_{j=1}^{3} Q_j = 0 \quad j = 1,2,3 \tag{11.12}$$

流量 Q_j 为正值表示水流向交叉点，Q_j 为负值表示水从交叉点流出。

各管道的流量由下式得出：

$$Q_j = 0.25\pi d_j^2 V_j s_j \quad j = 1,2,3$$

上式中，

$$V_j = \sqrt{\frac{2gd_j \mid \Delta h_j \mid}{\lambda_j L_j}} \quad \Delta h_j = h_j - h_p \quad j = 1,2,3$$

s_j 是表示 Δh_j 的符号，重力加速度 $g=9.81\ \mathrm{m/s^2}$，管道摩擦系数 λ_j 由式(11.11)得出，它是 R_{ej} 的函数，R_{ej} 表达式为[参见式(11.10)]：

① 参见 N.H.C.Hwang 和 C.E.Hita, *Fundamentals of Hydraulic Engineering Systems*，第二版，Prentice - Hall, Englewood Cliffs, NJ, 1987 年，第 106 ~ 110 页。

$$R_{ej} = \frac{V_j d_j}{v} \quad j = 1,2,3$$

式中,20℃时水的运动粘度 $v = 1.002 \times 10^{-6}$ m²/s。确定式(11.12)的 h_p 值。

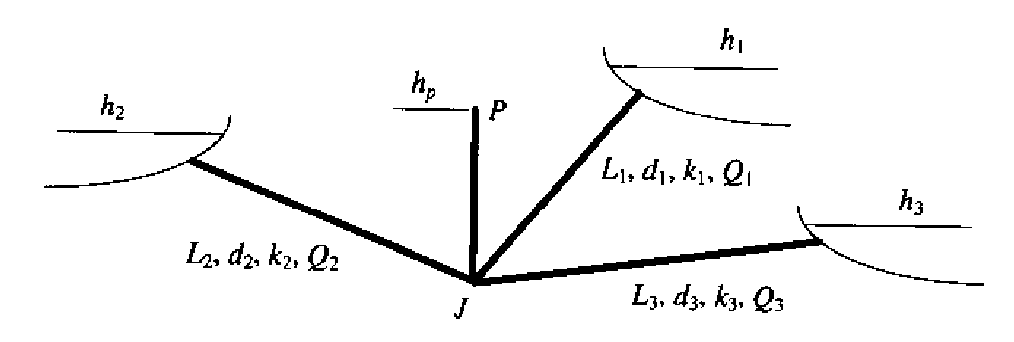


图 11.8　在 J 处连接三个水库的管道

求解方法如下:假定在 h_j 的允许范围内任取一点 h_p,当雷诺数 Re 非常大时由式(11.11)求出 λ_j 的假定值,利用这个假定值能计算出每个 V_j 值:

$$\lambda_j = \left[2\log_{10}\left(3.7\frac{d_j}{k_j}\right)\right]^{-2}$$

用 V_j 值求出 R_{ej} 值,然后根据通用的阔尔布鲁克公式[参见式(11.11)]再计算 λ_j 的值。重复该过程,直到 V_j 的值处于可接受范围内为止。

求出每个 V_j 后,计算 Q_j,并检验是否满足式(11.12)。如果不满足,则选择另一个 h_p 值,计算 V_j 的新值。应当注意当 $\Delta h_j = 0$ 时,$Q_j = 0$ 和 $R_{ej} = 0$,因此不能计算出 λ_j。在实现该过程的代码和函数中应用了 fzero 函数的嵌套,内部函数 fzero 用来求解 V_j,外部函数 fzero 求解 h_p。

根据上述求解方法,可求出在表 11.2 所示参数下的流速。

表 11.2　图 11.8 中水库的参数

水库	d_j(m)	L_j(m)	k_j(m)	h_j(m)
1	0.30	1 000	0 000 60	120
2	0.50	4 000	0 000 60	100
3	0.40	2 000	0 000 60	80

现创建三个函数,第一个函数为 ReservoirSumQ,用于求解 Q_j 和计算式(11.12);第二个函数为 PipeFrictionCoeff,用于计算式(11.11);第三个函数为 ResFriction,用于计算每个 V_j 值下的式(11.11)。下面分别给出这三个函数。函数 ReservoirSumQ 为:

```
function[sq,q] = ReservoirSumQ(hg,d,el,k,h)
cv = 2 * 9.81 * d./el;
ro = d/1.002e - 6;
dk = d./k;
qd = 0.25 * pi * d.^2;
frictguess = (2 * log10(3.7 * dk)).^ - 2;
hh = h - hg;
options = optimset('display','off');
for n = :length(d)
```

```
    if hh(n) = = 0
      q(n) = 0;
    else
      lambda = fzero('ResFriction',frictguess(n),options,dk(n),hh(n),cv(n),ro(n));
      q(n) = sign(hh(n)) * sqrt(cv(n) * abs(hh(n)/lambda) * qd(n));
    end
  end
  sq = sum(q);
```

函数 PipeFrictionCoeff 为：

```
function x = PipeFrictionCoeff(el,re,dk)
if dk > 1000001dk = = 0
  x = el - (2 * log10(re * sqrt(el)/2.51))^ - 2;
else
  x = el - (2 * log10(2.51/re/sqrt(el) + 0.27/dk))^ - 2;
end
```

函数 ResFriction 为：

```
function lamb = ResFriction(lambda,dk,dh,cv,ro)
ren = sqrt(cv * abs(dh)/lambda) * ro;
lamb = PipeFrictionCoeff(lambda,ren,dk);
```

其程序代码为：

```
d = [0.3 0.5 0.4];
el = [1000 4000 2000];
k = [0.6 0.6 0.6] * 1e - 3;
h = [120 100 80];
options = optimset('display','off');
hg = fzero('ReservoirSumQ',110,options,d,el,k,h);
[sq,q] = ReservoirSumQ(hg,d,el,k,h);
disp(['Elevation h-sub-p = 'num2str(hg)'m'])
disp(['Q1 = 'num2str(q(1))'m^3/s Q2 = 'num2str(q(2))'m^3/s Q3 = '...
num2str(q(3))'m^3/s'])
```

执行结果为：

```
Elevation h_sub_p = 98.904 m
Q1 = 0.16185 m^3/s  Q2 = 0.068728 m^3/s  Q3 = - 0.23058 m^3/s
```

11.3 外部流动

11.3.1 无限平面上的边界层由静止突然启动

研究在 x-z 无穷轴平面上 $y = 0$ 到 $y = h = 10.0$ cm 之间的一层液体。其中 $y = 0$ 处是刚性平板，$y = h$ 处是自由液面。刚开始时，刚性平板和流体处于静止状态。$t = 0$ 时，在 x 轴的正方向上平板瞬间加速到速度 U。导致流体的运动仅仅是 x 轴方向，并且是时间和 y 坐标的一个函数。可以求出 U 为 5.0 cm/s 和运动粘度 V_{ω} 为 1.0 cm^2/s 时的函数关系式 $u = u(y,t)$。

对 Navier-Stokes 方程中的 x 部分进行处理可得到这种结果，公式可化简为：

$$\frac{\partial u}{\partial t} = v_{vis}\frac{\partial^2 u}{\partial y^2} \tag{11.13}$$

初始条件是：

$$u(y,0) = 0$$

平板表面的边界条件是滞流条件，为：

$$u(0,t) = U$$

当流体自由面（$y = 10.0$ cm）的边界条件等于零剪切应力时为：

$$\left.\frac{\mathrm{d}u}{\mathrm{d}y}\right|_{y=10.0} = 0$$

可以用 pdetool 来求解这个问题，在 pdetool 窗口上用 Option 菜单中的 Axes Limits 创建一个窗口，其尺寸为：x 方向上 -1.5 到 1.5，y 方向上 0.0 到 10.0。然后创建一个矩形，其尺寸为：x 方向上 -0.25 到 0.25，y 方向上 0.0 到 10.0。用 PDE 工具箱在（x,y）域内求解方程，这个计算过程与 x 无关。如果 x 计算量与 y 的计算量相比非常小，那么计算时间将会缩短。

矩形的边界条件如下设定：矩形底部（平板的表面）的边界条件为 $u = U = 5.0(h = 1, r = 5)$，矩形顶部（液体的自由面）的边界条件为$\partial u/\partial x = 0(q = g = 0)$，该条件也适用于与 x 无关的矩形的两个垂直边，用改进两次的网格可以求得这个解。为了在 pdetool 中确定式(11.13)，利用 PDE 下拉菜单中 PDE Specification 窗口选择 Parabolic 选项。然后设置 $c = 1$、$a = 0$、$f = 0$ 和 $d = 1$，最后用 Solve 下拉菜单的 Parameters 选项来确定计算结果的显示时间，在 Time 区域上输入 0:0.5:10。解这个方程，在 MATLAB 命令窗口上输出结果 u 和网格参数 p、e 和 t。u 是一个（$n_m \times n_t$）的矩阵，其中 n_m 是网格点的个数，n_t 是求解次数，该例中 $n_t = 21$。以下的程序代码是在 $x = 0$ 处求解的数值，得出如图 11.9 的曲线，其中曲线之间的时间间隔 Δt 为 0.5 s。

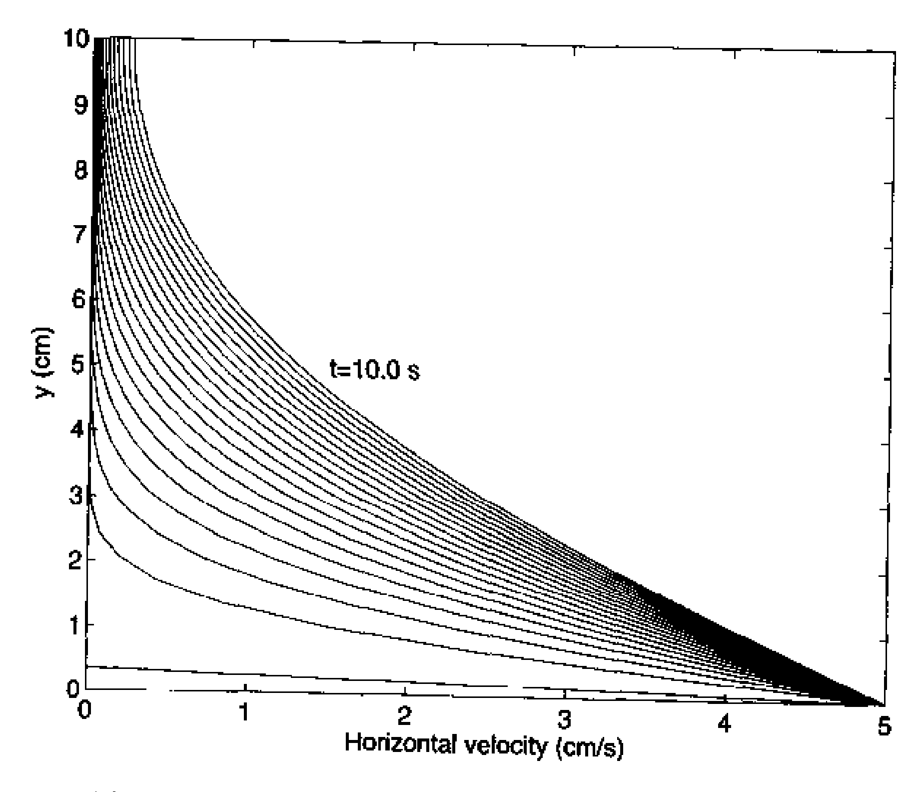


图 11.9　流层 10.0 cm 深处的水平速度突然加速到 $U = 5.0$ cm/s

```
ymin = 0.0;ymax = 10.0;
y = linspace(ymin,ymax,30);
[nm,nt] = size(u);
for i = 1:nt
  plot(tri2grid(p,t,u(:,i),0,y),y)
  hold on
end
axis([0,5,ymin,ymax])
ylabel('y(cm)')
xlabel('Horizontal velocity (cm/s)')
text(1.5,5,'t = 10 0s')
```

11.3.2 勃拉休斯边界层

在壁面的平面边界层上,如果是不可压缩流场,则其边界层方程为:

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0 \tag{11.14}$$

$$u\frac{\partial u}{\partial x}+v\frac{\partial v}{\partial y}=v_{w}\frac{\partial^2 u}{\partial y^2}$$

式中 x 是平行于物体表面的坐标,y 是垂直于物体表面的坐标,u 和 v 分别是相应的流速,V_w 是运动粘度。边界条件为平板表面上的 u 和 v 都为零,且当 $y\to\infty$ 时,$u\to U$,$v\to0$。

一个类似的解法是:

$$\frac{df}{d\eta}=\frac{u}{U} \tag{11.15a}$$

和

$$\eta=y\sqrt{\frac{U}{xv_w}} \tag{11.15b}$$

变量 f 与流函数成比例,$d^2f/d\eta^2$ 与切应力成比例。该解法可将边界层方程转换为一般非线性微分方程:

$$2\frac{d^3f}{d\eta^3}+f\frac{d^2f}{d\eta^2}=0 \tag{11.16}$$

式中当 $\eta=0$ 时,

$$f=0\quad \frac{df}{d\eta}=0 \tag{11.17a}$$

而当 $\eta\to\infty$ 时,

$$\frac{df}{d\eta}\to1 \tag{11.17b}$$

为求解式(11.16),引入下述定义将其简化成为三个一阶方程:

$$f_1=f$$

$$f_2=\frac{df}{d\eta} \tag{11.18}$$

$$f_3=\frac{d^2f}{d\eta^2}$$

得到:

$$\frac{df_1}{d\eta}=f_2$$
$$\frac{df_2}{d\eta}=f_3 \tag{11.19}$$
$$\frac{df_3}{d\eta}=-0.5f_1f_3$$

当 $\eta=0$ 时，边界条件变为：

$$f_1(0)=f_2(0)=0 \tag{11.20a}$$

而当 $\eta\to\infty$ 时，

$$f_2(\eta\to\infty)\to 1 \tag{11.20b}$$

用迭代法对给定方程组(11.20)的边界条件求式(11.19)的解。在这种方法中，$f_1(0)$和 $f_2(0)$由方程组(11.20a)给出，$f_3(0)$是壁面的剪应力的假定值。然后，用 ode45 对微分方程组(11.19)在外边界 $\eta=\eta_{max}$：$f_2(\eta_{max})$上积分得到 f_2。当 η_{max}足够大时[参见式(11.20b)]，$f_3(0)$和 $f_2(\eta_{max})$的精确值→1.0。在程序中 fzero 用迭代法求解精确的 $f_3(0)$值。ode45 调用函数 Blasius 对方程组(11.19)求解。Fzero 调用函数 Blasius2 执行迭代得到 $f_3(0)$的值，由这个值可以求出 $f_2(\eta_{max}\to\infty)\to1.0$，函数 Blasius2 也调用 Blasius。以下给出了这些函数。

Blasius 函数为：

```
function F = Blasius(x,y)
F = [y(2);y(3); -0.5 * y(1) * y(3)];
```

Blasius2 函数为：

```
function fn = Blasius2(fp0,EtaMax)
[eta ff] = ode45('Blasius',[0 EtaMax],[0 0 fp0]');
fn = 1.0 - ff(end,2);
```

程序代码为：

```
EtaMax = 20.0;
options = optimset('display','off');
shear0 = fzero('Blasius2',0.3,options,EtaMax);
disp(['The shear stress at eta = 0 is:' num2str(shear0)]);
[eta ff] = ode45('Blasius',[0 EtaMax],[0 0 shear0]');
plot(ff(:,1),eta,'k-',ff(:,2),eta,'k-.',ff(:,3),eta,'k--.')
axis([0 3 0 4])
ylabel('\eta')
xlabel('f,df/d\eta,d^2f/d\eta^2')
legend('f','df/d\eta','d^2f/d\eta^2',4)
```

执行这段程序得到在 $\eta=0$ 处的切应力为 0.332 03，结果如图 11.10 所示。

11.3.3 势流

对于不可压缩势流，速度场 $\vec{u}$ 受以下条件的限制：

$$\nabla\cdot\vec{u}=0$$

和

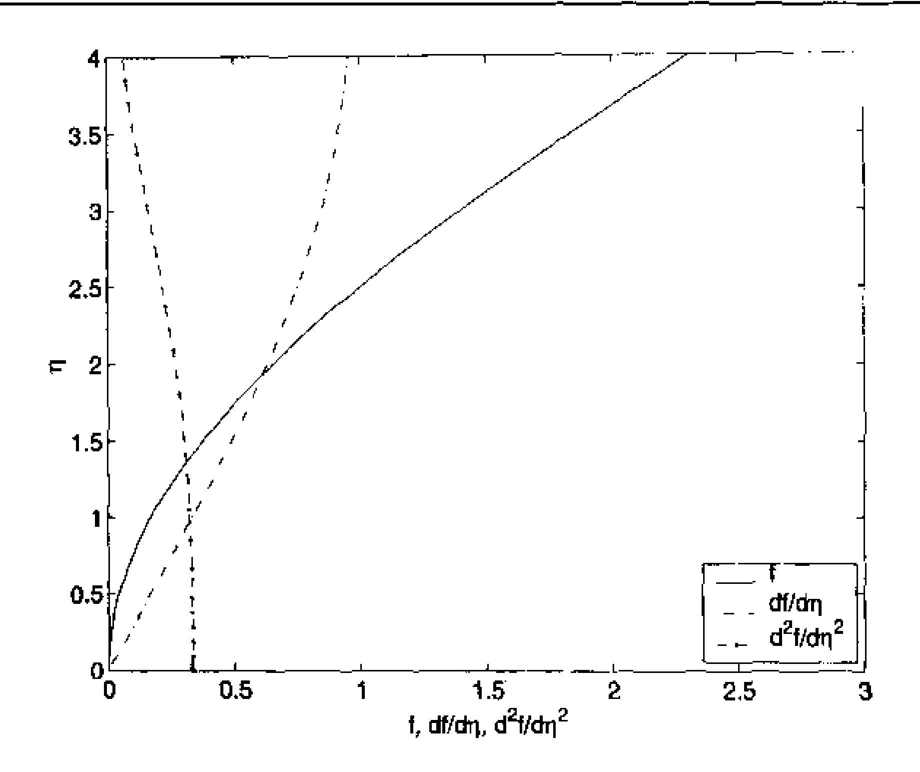


图 11.10 流函数 f_1、速度 f_2 的流向分量和切应力 f_3 的勃拉休斯边界层的剖面图

$$\nabla \times \vec{u} = 0$$

这些条件表明速度可表示为势场 ϕ 的梯度:

$$\vec{u} = \nabla\phi$$

式中的 ϕ 满足拉普拉斯方程:

$$\nabla^2\phi = 0 \qquad (11.21)$$

利用流函数可以得到二维空间流线簇的另一数学表达式,式中,

$$u = \frac{\partial \Psi}{\partial y}$$

$$v = -\frac{\partial \Psi}{\partial x}$$

流函数也满足拉普拉斯方程。边界条件可以由纽曼条件和第立克特条件组成,与边界正交的速度分量确定纽曼条件,ϕ 的值确定第立克特条件。固体边界处的纽曼条件是 $\vec{u}\cdot\hat{n} = 0$,式中 $\hat{n}$ 是边界的单位法向。在下文中将讨论几种获得二维空间流线簇流场的方法,这些方法中有两种是将已知的势函数或流函数叠加在一起来构建流线簇的。现给出这样的四个值。

点源和点汇:

$$\phi_M = \frac{m}{2\pi}\ln r_M \qquad \Psi_M = \frac{m}{2\pi}\theta_M$$

$$r_M^2 = (x - x_M)^2 + (y - y_M)^2 \qquad \theta_M = \arctan\frac{y - y_M}{x - x_M}$$

式中(x_M, y_M)是点源或点汇的坐标位置,m 是点源的强度。

偶极流:

$$\phi_K = \frac{K\cos\theta}{r_K} \qquad \Psi_K = -\frac{K\sin\theta}{r_K}$$

$$r_K^2 = (x - x_K)^2 + (y - y_K)^2 \qquad \theta_K = \arctan\frac{y - y_K}{x - x_K}$$

式中(x_K, y_K)是偶极流的坐标，K是偶极流的强度。

点涡：

$$\phi_\Gamma = \frac{\Gamma}{2\pi}\theta_\Gamma \qquad \Psi_\Gamma = -\frac{\Gamma\ln r_\Gamma}{2\pi}$$

$$r_\Gamma^2 = (x - x_\Gamma)^2 + (y - y_\Gamma)^2 \qquad \theta_\Gamma = \arctan\frac{y - y_\Gamma}{x - x_\Gamma}$$

式中(x_Γ, y_Γ)是旋涡的坐标位置，Γ是旋涡的强度。

均匀流场：

$$\phi_U = Ux \qquad \Psi_U = U_y$$

式中 U 是流速。

总之，可以把这些不同的流函数结合在一起模拟不同形状的流。如果 Ψ_s 是新的流线函数，则：

$$\Psi_s = \Psi_M + \Psi_K + \Psi_\Gamma + \Psi_U$$

方法一：用 contour 绘出流线模型

第一种也是最简单的确定流型方法是用 contour 绘制流线。以下的程序代码画出了一种流体的流线图。这种流体由均匀流、偶极流和点涡组成。其中，均匀流速度为 U，偶极流的坐标为(x_K, y_K)且强度为 K，点涡的坐标为(x_Γ, y_Γ)且强度为 Γ。为了绘出该结果，选择偶极流和点涡的坐标为$(-1, -1)$，每个量的强度数值为：

$$K = 5.0 \qquad (x_K, y_K) = (-1, -1)$$

$$\Gamma = 8\pi \qquad (x_\Gamma, y_\Gamma) = (-1, -1)$$

$$U = 5.0$$

如图 11.11 所示，流线是沿着一个圆柱体流动的。绘出该图的主要难点是：如何选择等高线以得到完整描述轮廓的基准面。可以用区域左下角的 Ψ_s 值作为最小值，用区域顶部中间的 Ψ_s 值作为最大值来实现。在图 11.11 中圆柱体的表面（同时也是一条流线）已经叠加到这些流线上了。

```
nx = 100; xmin = -3.5; xmax = 1.5;
ny = 100; ymin = -3.0; ymax = 1.5;
[x y] = meshgrid(linspace(xmin, xmax, nx), linspace(ymin, ymax, ny));
U0 = 5.0;
Gamma = 8 * pi; xGamma = -1.0; yGamma = -1.0;
K = 5.0; xK = -1.0; yK = -1.0;
radius = inline('sqrt((x - x1).^2 + (y - y1).^2)', 'x', 'y', 'x1', 'y1');
PsiK = K * sin(atan2(y - yK, x - xK))./radius(x, y, xK, yK);
PsiGamma = Gamma * log(radius(x, y, xGamma, yGamma))/2/pi;
StreamFunction = U0 * y - PsiGamma - PsiK;
levmin = StreamFunction(1, nx);
levmax = StreamFunction(ny, nx/2);
levels = linspace(levmin, levmax, 50)';
contour(x, y, StreamFunction, levels)
hold on
```

```
theta = linspace(0,2 * pi);
plot(xGamma + cos(theta),yGamma + sin(theta),'k')
axis equal
axis([xmin xmax ymin ymax])
ylabel('y')
xlabel('x')
```

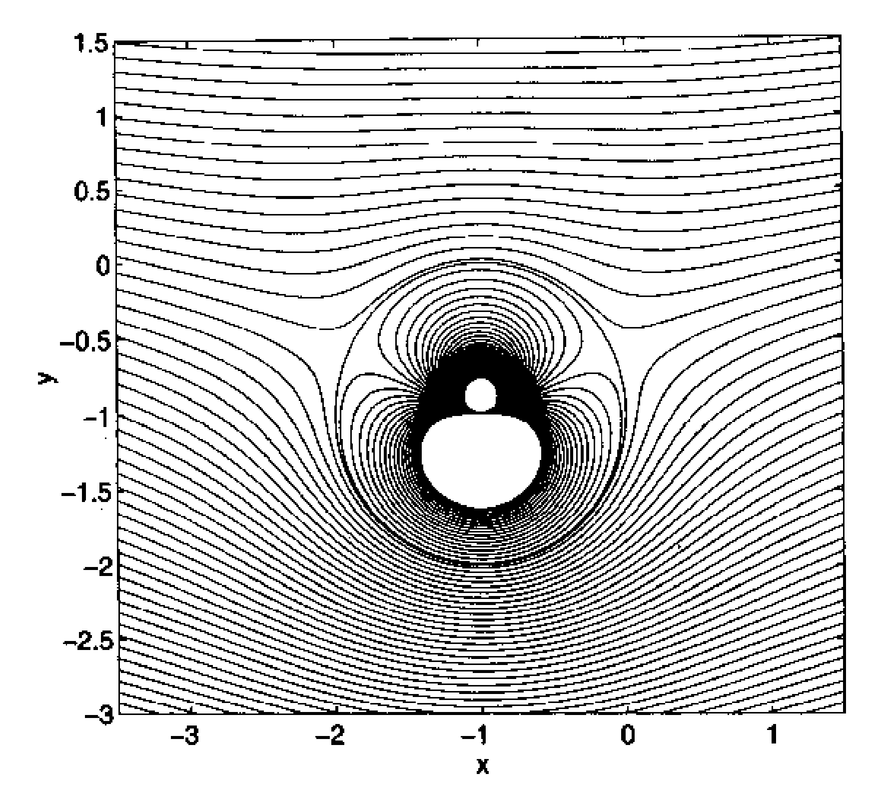


图 11.11 从 Ψs 的轮廓线得到交叉流动中环绕圆柱体的流线

方法二:直接计算流线

第二种确定流型的方法是用 fzero 确定流线。举一个例子:假定流是由均匀流和点源组成,均匀流在 y 轴正方向上 $U=1$,一个强度 $m=4.0$ 的点源在(0, -1)处,另一个强度 $m=-4.0$ 的点源在(0,1) 处。则有:

$$\Psi_{total} = +\Psi_U + \Psi_{M_1} + \Psi M_2$$

这些分量可以得出一个椭圆形的均匀流,椭圆方程由下式给出①:

$$\frac{2x}{x^2+y^2-a^2} = \tan\frac{xU}{m/2\pi} \qquad (11.22)$$

式中 U 是流速,m 是点源强度,a 是特征尺寸。

在这个例子中使用 fzero 的困难是如何选择一个恰当的初始值。在下面的程序代码中,通过在直线 $y=-2.0a$ 上的一系列 x 点找到流函数 Ψ 的值来求解这个问题。其中给定初始值,在 $y=-2.0a$ 条件下绘制流线,在每个连续的 y 点上,fzero 用 StreamFun 函数求出流函数中的 x 点的坐标。x 值作为假设值对应前面的y 值。假定 $a=1$ 时程序输出如图 11.12 所示。这个图旋转 90°就得到了习惯上的水平流。因为它是由式(11.12)直接画出来的,所以这里并未得

① 参见 L.M. Milne-Thomson, *Theoretical Hydrodynamics*, Dover, Mineola, NY, 1996 年,第 216 页。

出与椭圆的边界相一致的流线。程序代码为：

```
U = 1.0;a = 1.0;m = 4.0;co = m/(2 * pi);
nPsi = 15;n = 30;yStart = -2.0 * a;
xStart = linspace(0,2 * a,nPsi);
y = linspace(-2 * a,2 * a,n);
x = zeros(1,n);
StreamFun = inline ('-U * x - co * (atan2(x,y + a) - atan2(x,y - a)) - psi',...
                    'x','y','psi','U','co','a');
Psi = StreamFun(xStart,yStart,0,U,co,a);
options = optimset('display','off');
for j = 1:nPsi
  guess = xStart(j);
  for i = 1:n
    x(i) = fzero(StreamFun,guess,options,y(i),Psi(j),U,co,a);
    guess = x(i);
  end
  if j > 1
    plot(y,x,'b',y,-x,'b')
  end
  hold on
end
axis([-2 * a,2 * a,-2 * a,2 * a])
ylabel('x')
xlabel('y')
xx = linspace(-1,1,40);
yy = sqrt(1 - xx.^2 + 2 * xx./tan(xx/co));
plot(yy,xx,'k',-yy,xx,'k')
```

其中 *nPsi* 是流线的条数，*n* 是沿每条流线计算出的点数。

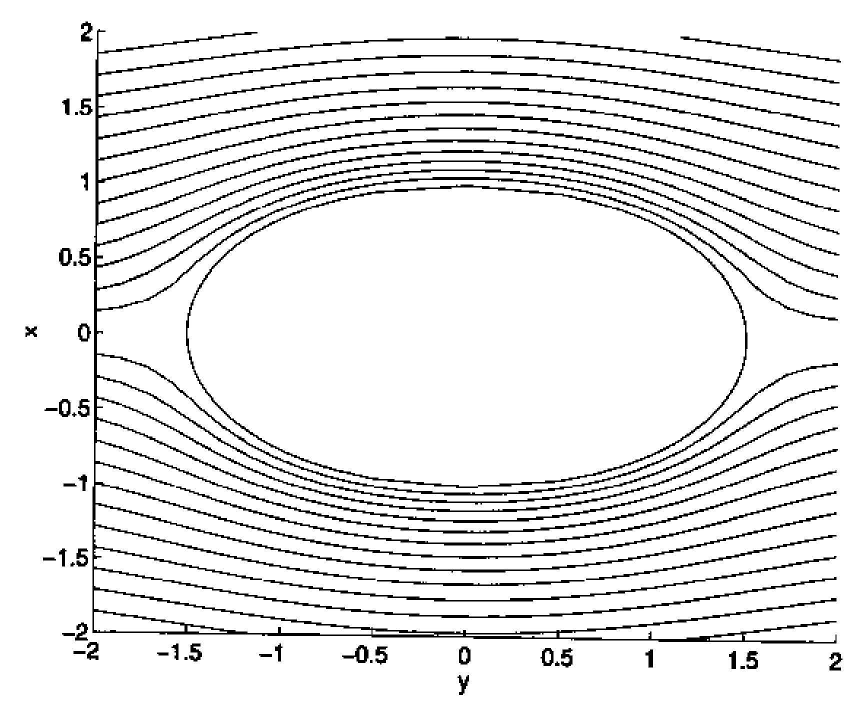


图 11.12　式(11.22)表示的椭圆流线

方法三：用 pdetool 求解流场

第三种方法是用 pdetool 直接对式(11 21)求解，解决势流问题。一个半径为 1.0 的圆柱体放置在流速为 10.0 的管道中央，现求解该圆柱体的流场问题。在 pdetool 窗口上选定窗口的轴线为：在 x 方向上从 -3.5 到 3.5，在 y 方向上从 -2.5 到 2.5，然后以(0,0)为中心点、半径为 1.0 做一个圆($C1$)；以(0,0)为中心、长度为 6.0、高度为 5.0 做一个矩形($R1$)。改变 Set Formula 使之成为 $R1-C1$，然后选择 Boundary Mode。

选定 Specify Boundary Conditions，在圆上和矩形的上下边界上选定纽曼边界条件 $\hat{n}\cdot\nabla\phi=0$。为实现该条件，设定 $c=1, q=0$ 和 $g=0$。在矩形的左边界上选定纽曼边界条件 $\hat{n}\cdot\nabla\phi=10$，即设定 $c=1, q=0$ 和 $g=10$。在矩形的右边界上选定纽曼边界条件 $\hat{n}\cdot\nabla\phi=-10$，即设定 $c=1$，$q=0$ 和 $g=-10$。因为在 pdetool 中边界的单位正交 $\hat{n}$ 从边界指向流体区域，所以前面的条件表示流是从左到右的。

接下来初始化网格并对其做两次改进。在 PDE 下拉菜单的 PDE Specification 上选择 elliptic 来确定偏微分方程，方程为 $\nabla^2 u=0$，该例中 MATLAB 变量 u 代表 ϕ；为此，设定 $c=1, a=0$ 和 $f=0$。求得 u 并在 u 的等高线上绘出速度 ∇u 的方向，如图 11.13 所示。

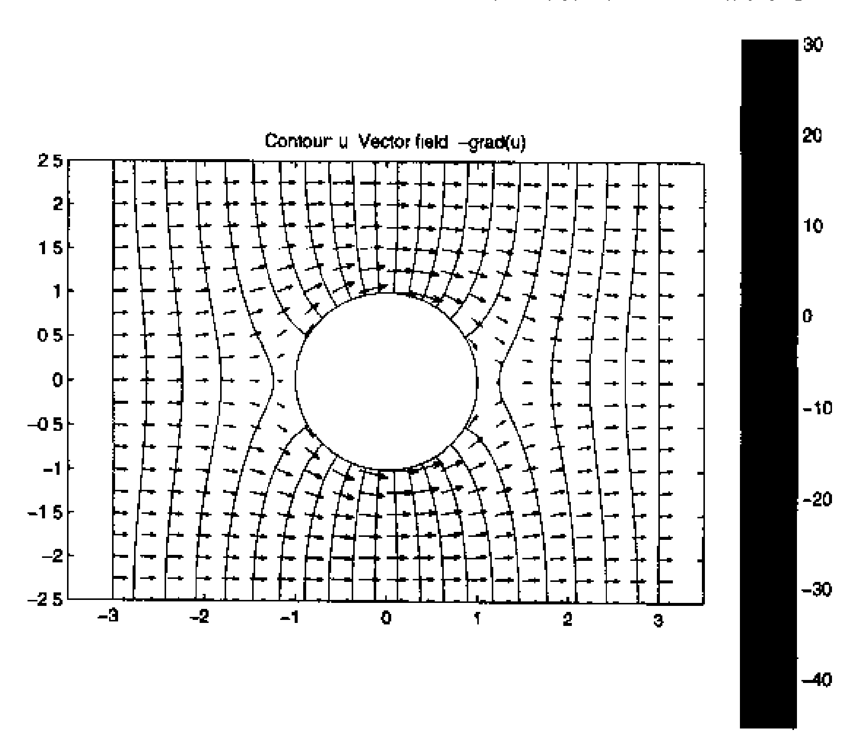


图 11.13 对式(11.21)求解得到的速度值(等高线)和方向

为得到更详尽的结果，把 u 的结果和网格坐标 p、e 和 t 的值输出到 MATLAB 命令窗口上。沿着任意一条垂直或水平线的速度曲线用下面的方法都可以得出。分析沿着圆柱体顶部到矩形顶部垂线 $x=0$ 的水平速度，用以下程序可以绘出速度分布的曲线图。在程序代码中首先创建一个矩形表格，使用 tri2grid 设定 $-0.5\leqslant x\leqslant 0.5$($n_x=9$ 点)和 $0.5\leqslant y\leqslant 2.5$($n_y=2.5$ 点)，然后用 gradient 函数得到不同区域。最后，通过在 x 方向分隔网格划分适当的差值，从而在线 $x=0$ 上得到梯度的水平分量。在程序中速度的结果数组称为 ux，图 11.14 给出 ux 与 y 的关系曲线图。对于无限流场中的一个圆柱体，最大速度为 $2U$，出现的位置在圆柱体的顶部和底

部(相对于流向 ± 90°)。在该例中,将圆柱放在由矩形确定的管道中,圆柱体上最大速度再次出现在 90°处,这个值为 $ux(1)=2.264U$。

```
nx = 9; xmin = -0.5; xmax = 0.5;
x = linspace(xmin, xmax, nx);
ny = 25; ymin = 0.5; ymax = 2.5;
y = linspace(ymin, ymax, ny);
uxy = tri2grid(p,t,u,x,y);
[DX,DY] = gradient(uxy);
ux = -DX(:,(nx-1)/2)/((xmax-xmin)/(nx-1));
plot(ux,y)
axis([10,25,0.9,2.5])
ylabel('y')
xlabel('Horizontal velocity')
hold on
plot([10,25],[1,1])
text(10.5,1.05,'Top surface of cylinder')
ux1 = max(ux)
```

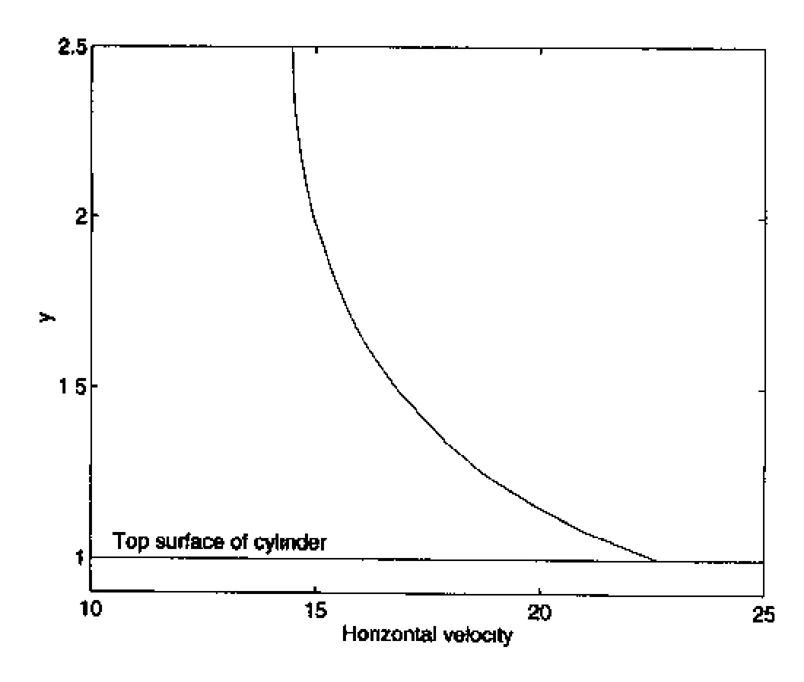


图 11.14　沿圆柱体顶面垂线方向的水平速度分布

练习

11.1 计算具有相同横截面积的两条管道中区域的流场。其中一条管道的截面为正方形,另一条管道的截面为矩形。矩形截面的一条边长是另一条边的四倍。用 pdetool 函数比较两管道体积流量。控制方程由式(11.7)给出。设式(11.7)的右边对于每条管道都是相同的,定为 1.0,即它们有相同的流体和压强梯度。管壁的边界条件为 $u=0$,在 MATLAB 命令窗口中输出每个结果,用第 11.2.1 节中的求解过程可以得出:

$$Q_{sq}/Q_{rect}\approx 2$$

式中 Q 为流量。

11.2 对称的薄机翼的流动可近似用势流理论表示①。翼弦在 x 轴上从 $x=0$ 延伸到 $x=c$，且由强度为 $\gamma(x)$ 的旋涡层表示。$\gamma(x)$ 为：

$$\gamma(\theta) = 2\alpha V_\infty \frac{1+\cos\theta}{\sin\theta}$$

式中，

$$x = \frac{c}{2}(1-\cos\theta) \qquad 0 \leqslant \theta \leqslant \pi$$

α 是入流相对于 x 轴的冲角(单位为弧度)，V_∞ 是流速。研究由 N 个离散旋涡近似组成的旋涡层，这个离散旋涡的间隔 $\Delta x = c/N$，强度 $\Gamma_i = \gamma(\theta_i)\Delta x$。用第 11.3.3 节的第一种方法，绘出 $\alpha = 10°$，$c = 2$ m 和 $V_\infty = 100$ m/s 的流线。其结果应与图 11.15 所示相同。

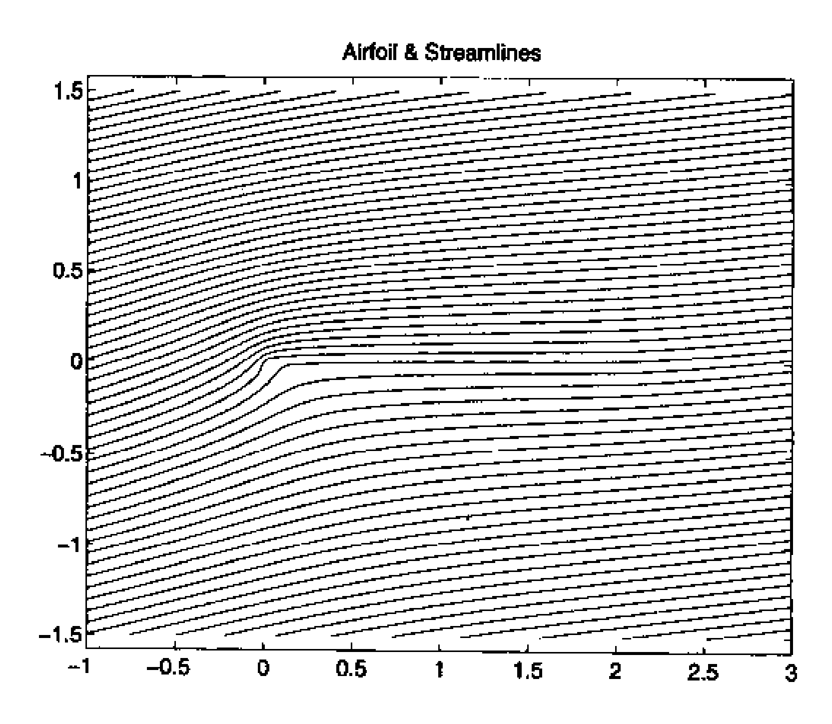


图 11.15 翼线 2 米、冲角 10 度的薄机翼流线，流向从左至右，机翼从(0,0)延伸到(2,0)

11.3 研究管道中绕圆柱体的流场，如图 11.16 所示。假定一势流，用 pdetool 绘出流线。把网格和结果变量输出到 MATLAB 的命令窗口中，计算沿管道底面的速度分布。其结果应与图 11.17 所示相同。

11.4 研究通过在平面壁附近的圆柱体的势流，如图 11.18 所示。已知流以 $U = 1.0$ m/s 的均匀流速通过这个圆柱体和平面壁，两偶极流的中心点坐标(x, y)分别为$(0, 0.25D)$和$(0, -0.25D)$，每个偶极流的速度势为：

$$\phi = \frac{UD^2}{4r}\cos\theta$$

式中 r 和 θ 是柱体的极坐标，坐标原点在偶极流的中心，绘出流线的形状。注意，环

① 参见 J. D. Anderson, *Fundamentals of Aerodynamics*, McGraw-Hill, New York, 1991 年，第 4 章。

绕着每个偶极流的闭合流线并不是圆形。绘出平面壁附近的速度分布情况，然后与练习11.3的结果相比较。

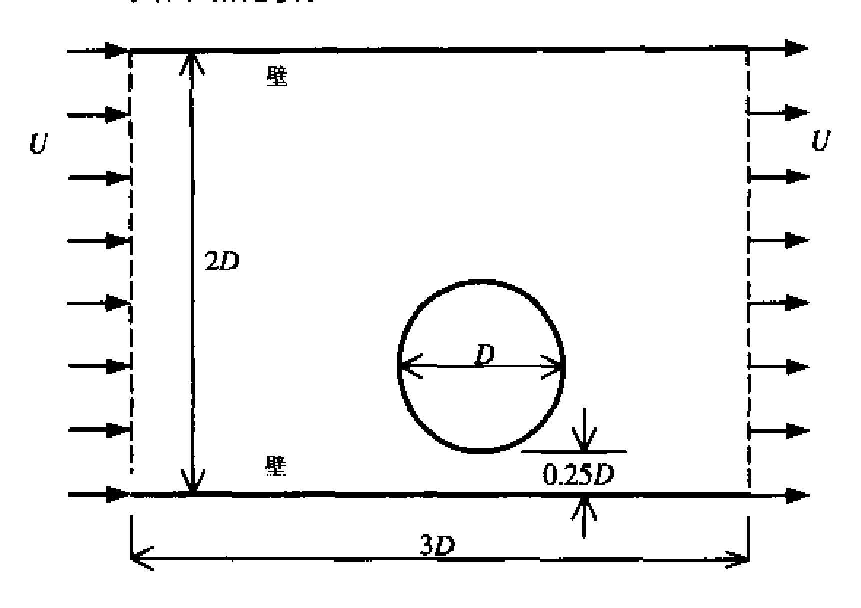


图11.16 水流以均匀水平流速 $U=1$ m/s，从左侧流入管道，从右侧流出管道，柱体直径 $D=1.0$ m，中心位置坐标 $(x,y)=(0.0,0.75)$ m

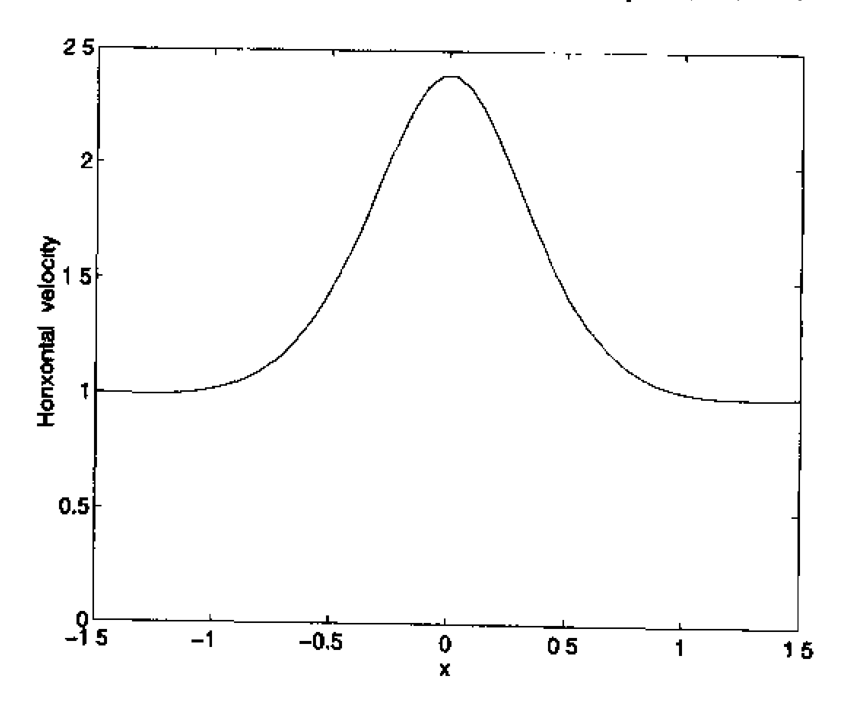


图11.17 沿管道底面壁的水平流速

11.5 在一条高 $2h$ 的水道中通过压强作用的平稳层流，如图11.19所示。速度由下式给定：

$$u(y)=-\frac{h^2}{2\mu}\frac{dP}{dx}\left(1-\frac{y^2}{h^2}\right)$$

式中 dP/dx 是压强梯度，μ 是流的动力粘度。用 pdetool 对该速度分布和另一个管道的速度分布进行比较，其中管道的高度与水道相同，宽度是水道高度的两倍。假设有相同的压强梯度和速度。求解的微分方程由式(11.7)给定。其结果应与图11.20所示相同。

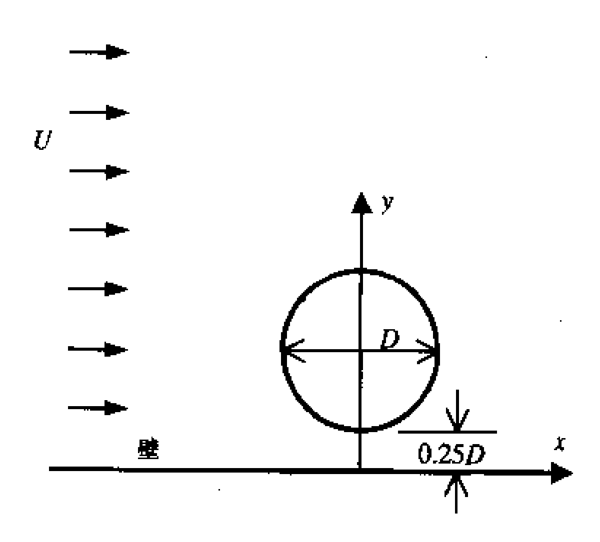


图 11.18 均匀水平流通过靠近平面壁的圆柱体

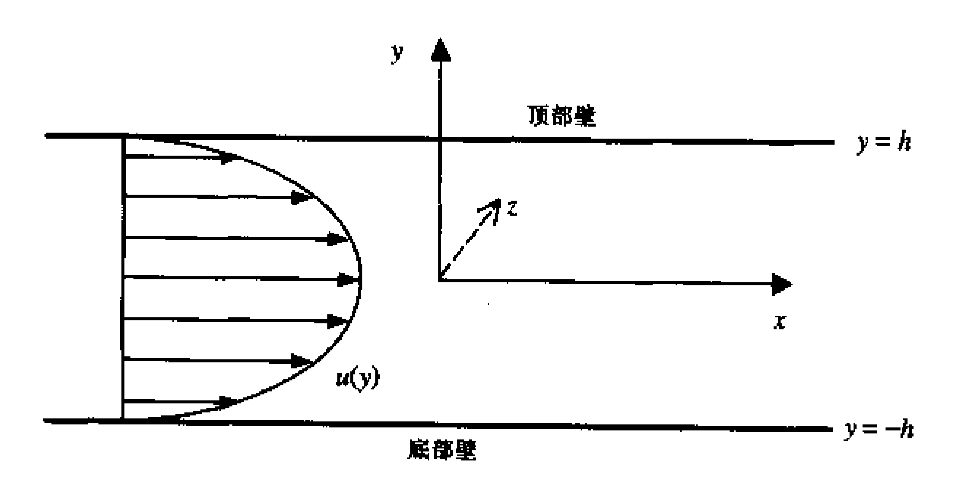


图 11.19 水道沿 z 轴无限延伸(对于管道,在 $z=\pm 2h$ 处放置两个垂直壁)

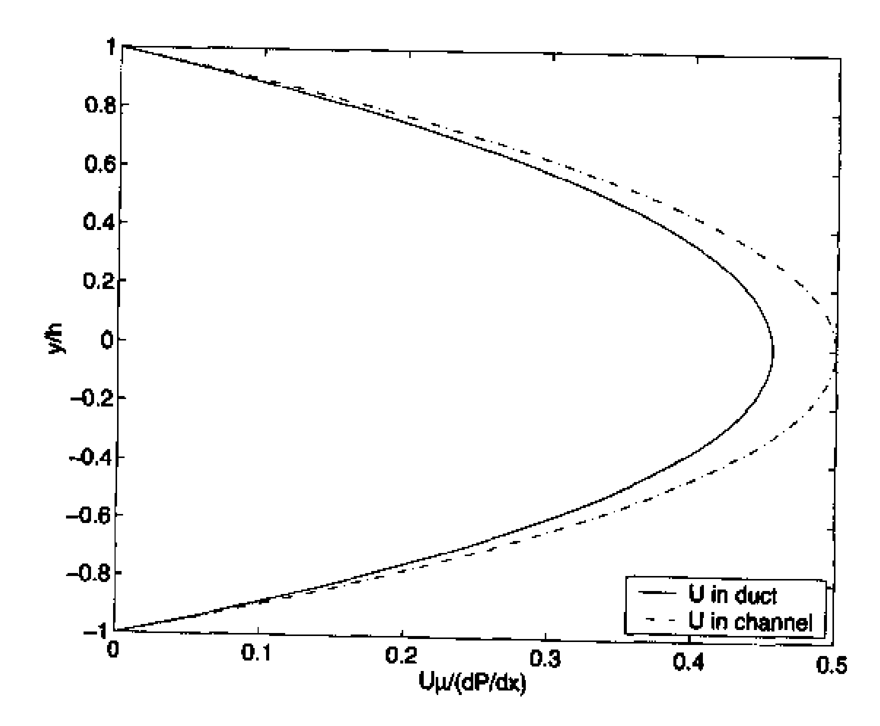


图 11.20 沿管道中心线的速度分布与同样高度水道中速度分布的比较

11.6 研究在一条管道中层流压强驱动充分发展且平稳的油的流动情况，如图11.21所示。该管道的横截面为矩形，其高为1 cm，宽为0.5 cm。油在压强梯度（在图11.21中，方向指向书页外面）作用下流动，在右壁以0.5 m/s的速度沿流动方向移动，而其他三个平面壁静止不动。假定，压强梯度为10 kPa/m，油的动力粘度为0.1 kg/m·s。用pdetool绘出管道中心处水平和垂直平面的速度分布情况。求解的微分方程由式(11.7)给定。其结果应与图11.22和图11.23一致。

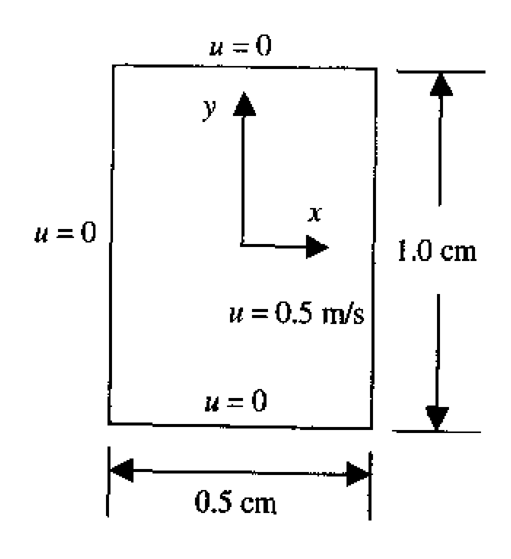


图11.21 练习11.6中管道的横截面

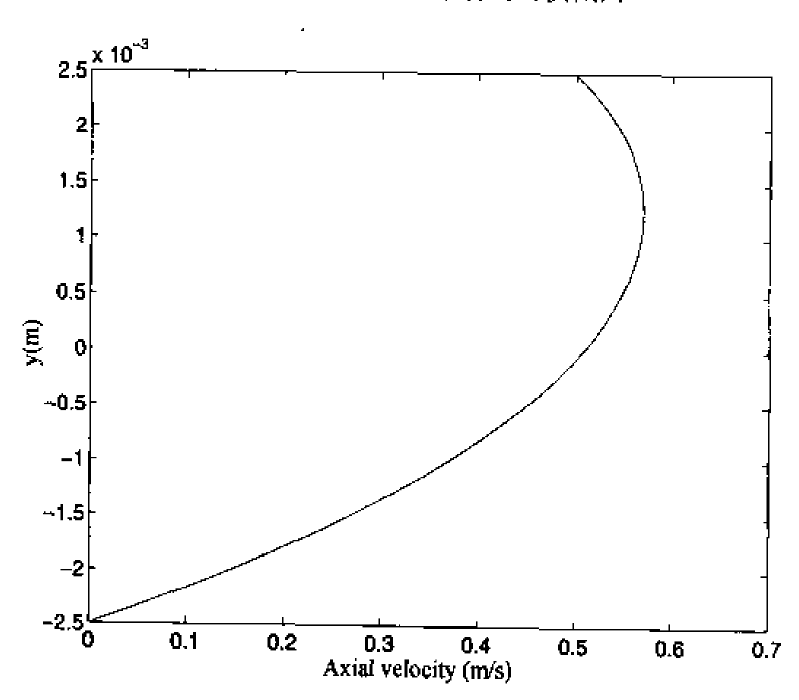


图11.22 连接固定壁和1 cm长移动壁的中心面轴向速度

11.7 分析在长和宽分别为0.3 cm和0.2 cm的矩形管道中的一种粘性流体，它的 μ = 0.02 kg/m·s，ρ = 800 kg/m^3。开始时流体静止，当 t = 0时，管道轴向压强梯度突然达到10 kPa/m。在0到0.04 s时间段内（步长为0.005 s），绘出连接管道平面壁0.3 cm长的中心平面的速度等高线图形。其结果应与图11.24所示一致。

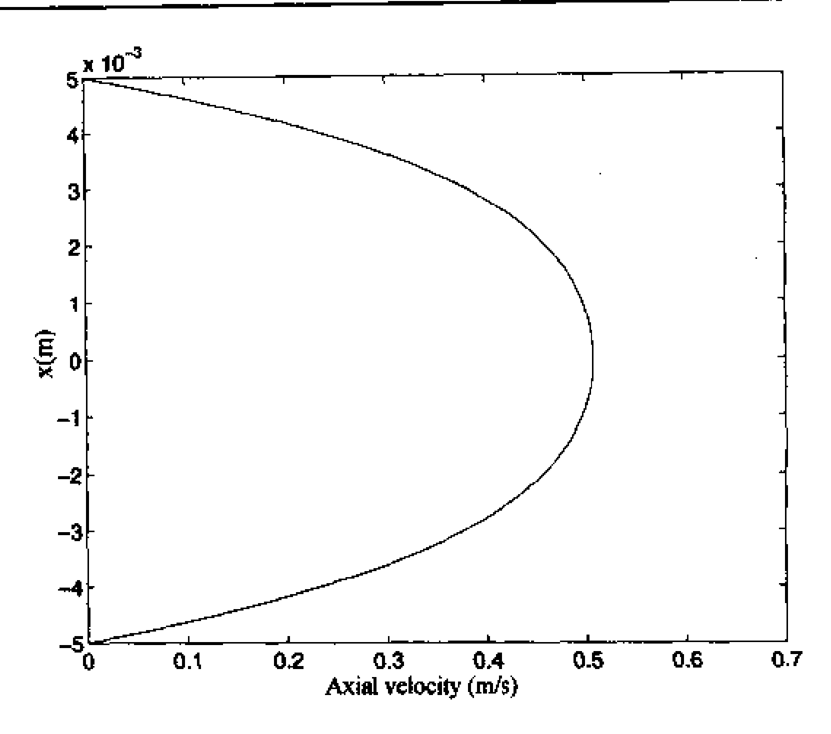


图 11.23 连接两个 0.5 cm 长固定壁的中心面轴向速度

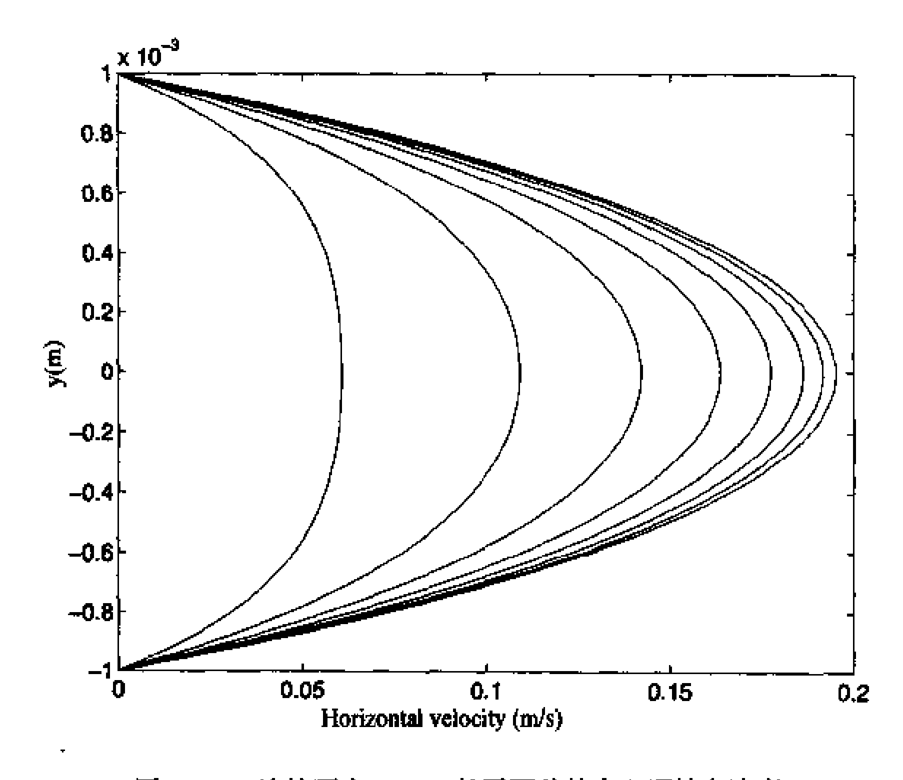


图 11.24 连接两个 0.3 cm 长平面壁的中心面轴向速度

11.8 研究通过一条截面为矩形的无缝管道的粘性流体，管道的矩形截面长和宽都是 a。当时间 $t=0$ 时，流体以 1 m/s 均匀速度通过截面。当 $t>0$ 时，应用壁的滞流边界条件和轴向压强梯度 dP/dx。[①] 这样，经过长时间后的稳定流量 Q_f 就等于 $t=0$ 时刻的流量 Q_0：

① 参见 F. M. White, *Fluid Mechanics*，第四版，McGraw-Hill，New York，1999 年，第 365 页。

$$\frac{\mathrm{d}P}{\mathrm{d}x}=\frac{28.46\mu Q_0}{a^4}$$

式中 μ 是流体的动力粘度。假定 $a=1$ cm，流体是 $\mu=0.2$ kg/m·s，$\rho=800$ kg/m^3 的油。利用从 pdetool 得到的结果绘出各个时刻通过管道中心面轴向流速的分布情况，以及管道中流量与时间的关系图。其结果应分别对应于图 11.25 和图 11.26。

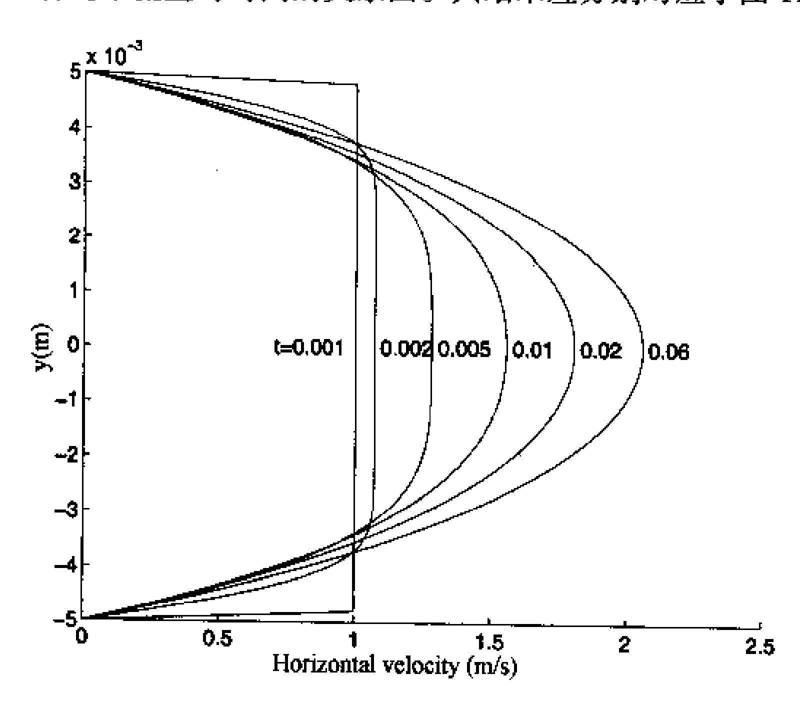


图 11.25　各个时刻通过管道中心面的轴向速度分布

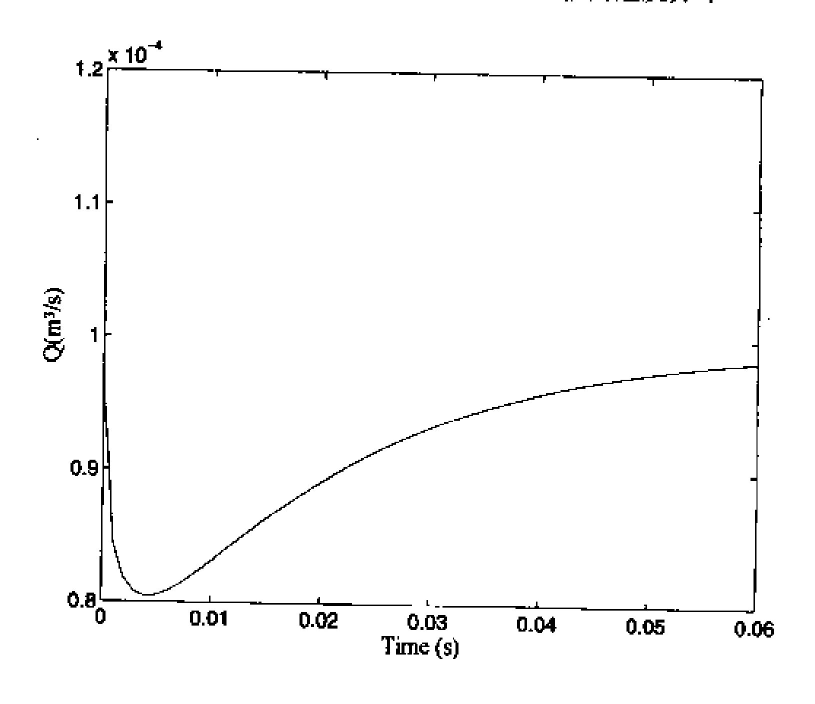


图 11.26　管道中流速与时间的关系

11.9　设定水是通过管道系统从水库 A 流向水库 B 的，如图 11.27 所示。当闸门完全打开时，流量为 0.003 m^3/s。常用的水头损失方程为：

$$\frac{P_1}{\rho g}+\frac{V_1^2}{2g}+z_1=\frac{P_2}{\rho g}+\frac{V_2^2}{2g}+z_2+\frac{\lambda LV^2}{2gD}+\sum_{m=1}^{5}K_{Lm}\frac{V^2}{2g}$$

式中，K_{Lm}是在这些区域(如图 11.27 所示)中的局部损失系数，λ 是管道的沿程损失系数，$V=4Q/\pi D^2$ 是管道中的平均速度。已知 $v_{vis}=1.3\times10^{-6}\ \mathrm{m^2/s}$ 和 $\rho=1\ 000\ \mathrm{kg/m^3}$，求管道的直径。

[答案：$D=0.051$ m。]

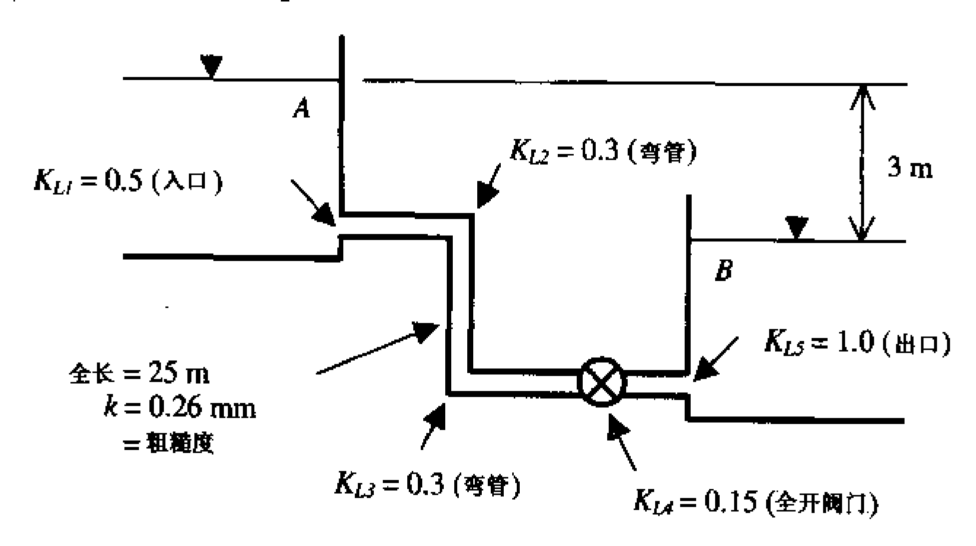


图 11.27　两水库之间的管道系统

11.10 图 11.28 所示的薄翼型流可以用势流理论近似。① 翼弦从 x 轴的 $x=0$ 延伸到 $x=c$，并且用沿翼弦的旋涡线来表示。旋涡线的强度 $\gamma(x)$为：

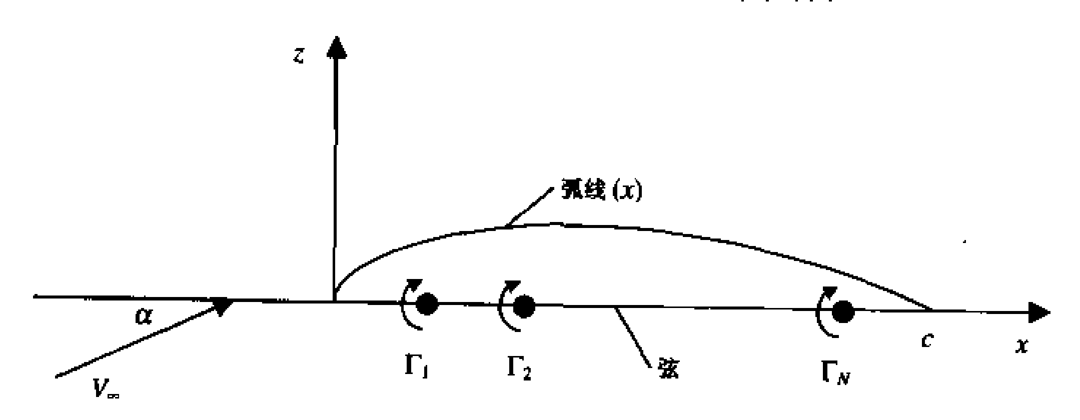


图 11.28　弦线上一个旋涡薄片的放置情况

$$\gamma(\theta)=2V_\infty\left(A_0\frac{1+\cos\theta}{\sin\theta}+\sum_{n=1}^{\infty}A_n\sin(n\theta)\right)$$

式中，

$$\frac{x}{c}=\frac{1}{2}(1-\cos\theta)\qquad 0\leqslant\theta\leqslant\pi$$

α 是入流方向相对 x 轴的冲角(弧度表示)，V_∞是流速。常数 A_n 为：

① 参见 J.D.Anderson，出处同上。

$$A_0 = \alpha - \frac{1}{\pi}\int_0^{\pi}\frac{dz}{dx}d\theta_0$$

$$A_n = \frac{2}{\pi}\int_0^{\pi}\frac{dz}{dx}\cos(n\theta_0)d\theta_0$$

其中 $z(x)$是翼弦到弧线的垂直距离。

令弧线为

$$\frac{z}{c} = 2.6595\frac{x}{c}\left[\left(\frac{x}{c}\right)^2 - 0.6075\frac{x}{c} + 0.1147\right] \quad 0 \leqslant x/c \leqslant 0.2025$$

$$= 0.02208\left(1 - \frac{x}{c}\right) \quad 0.2025 \leqslant x/c \leqslant 1.0$$

其中 z 是对翼弦的法向距离。当 $n = 0,1,\cdots,20$ 时,求 A_n 的值。如练习 11.2,在 $0 \leqslant x/c \leqslant 1$ 范围内,令沿着 x 轴的旋涡线近似用一系列离散的旋涡值 N_v 来表示。旋涡由距离 $\Delta x = c/N$ 分割,拥有的强度 $\Gamma_i = \gamma(\theta_i)\Delta x$。使用 contour 绘制出 $\alpha = 10°$, $c = 2$ m 和 $V_\infty = 100$ m/s 时的流线图。结果如图 11.29 所示。

[部分答案:$A_0 = 0.0412, A_1 = 0.0955, A_2 = 0.0792, A_3 = 0.0568$。]

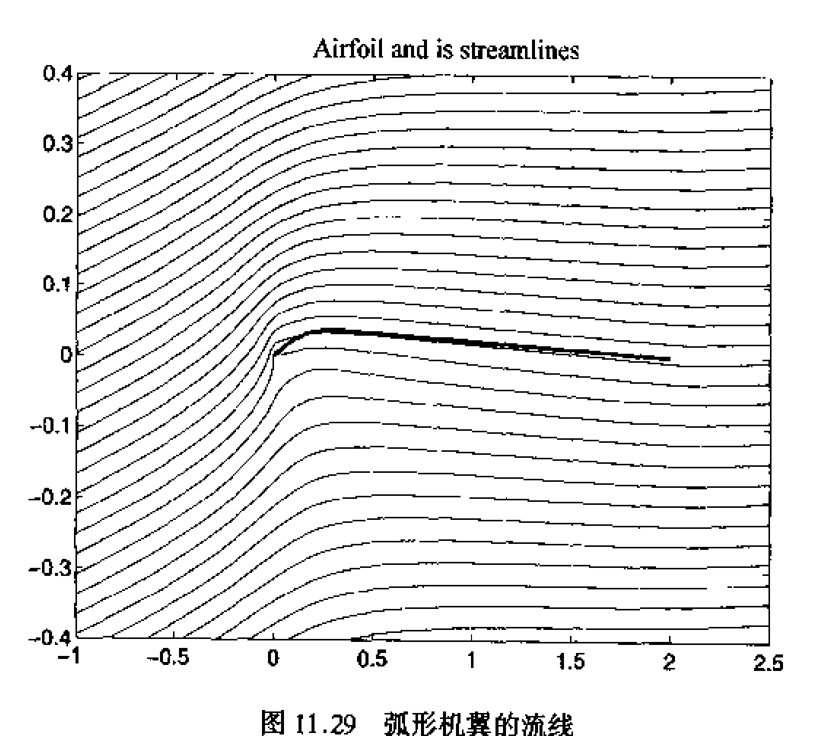


图 11.29 弧形机翼的流线

11.11 两个截面为矩形,面积相等的水库,由长为 L 的长管连接,高度差为 Z,如图 11.30 所示。从高度 Z 突然释放的水产生的振动可以由下式决定:①

$$\frac{d^2Z}{dt^2} + \text{signum}(dZ/dt)p\left(\frac{dZ}{dt}\right) + qZ = 0$$

式中,

① 参见 D.N.Roy, *Applied Fluid Mechanics*, Ellis Horwood Limited, Chichester, England, 1988 年,第 290~293 页。

$$p=\frac{fA_1A_2L_e}{2DaL(A_1+A_2)}\qquad q=\frac{ga(A_1+A_2)}{A_1A_2L}$$

假设液体为紊流，以使水头损失与速度平方成比例。数值 L_e 等价于管道局部损失的长度，g 是重力加速度，f 是管子的摩擦系数，A_1 和 A_2 是两个水库的表面积，a 是管子的截面积，D 是直径。

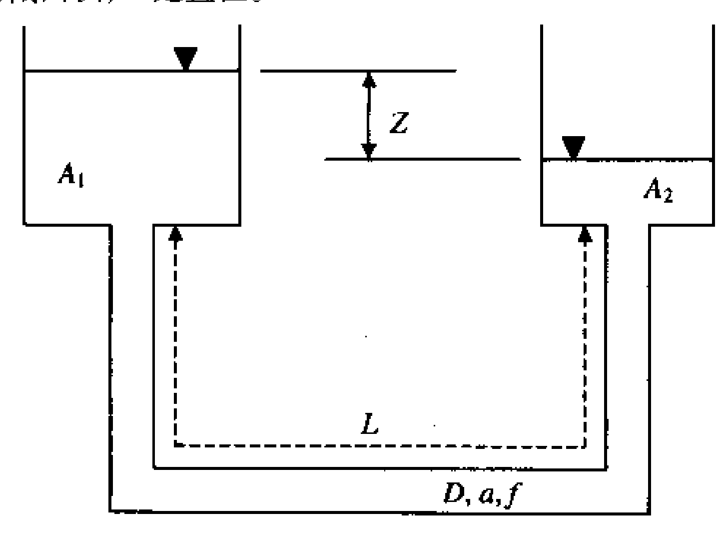


图 11.30 连通的水库

如果 $p=0.375\ \mathrm{m}^{-1}$，$q=7.4\times10^{-4}\ \mathrm{s}^{-2}$，并且初始条件 $Z(0)=Z_n$ m，$dZ(0)/dt=0$ m/s，当 $Z(t_n)=0$，$Z_n=5,10,\cdots,50$ 时，试确定第一次出现的 t_n 的值并绘制结果曲线，曲线应如图 11.31 所示，可使用 interp1 函数求解 t_n。

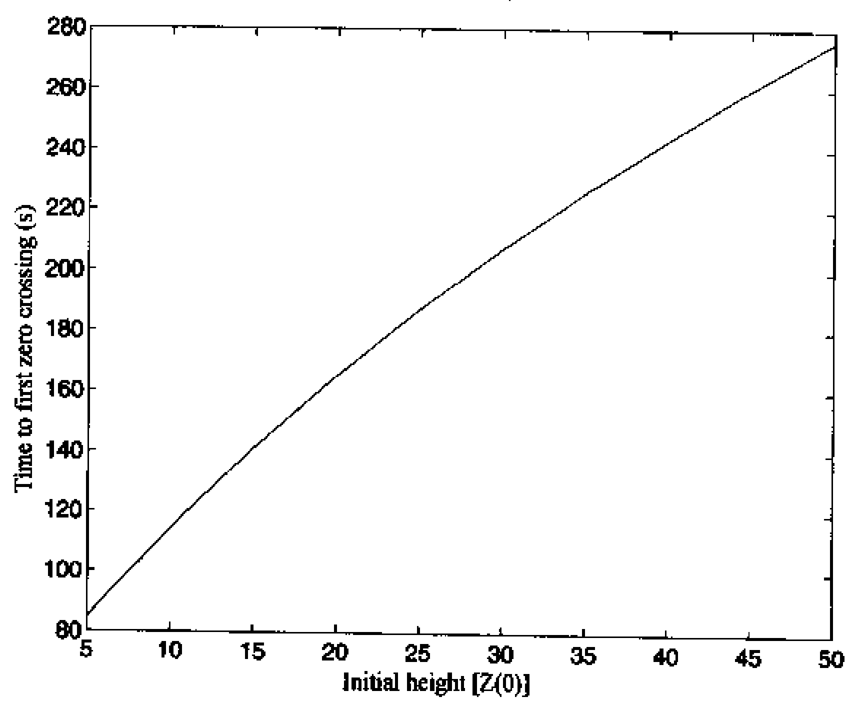


图 11.31 当 $Z(t_n)=0$ 时第一次得到的 t_n 值与 $Z(0)=Z_n$ 的函数关系

参考文献

1. Munson B R, D F Young, and Okiishi T H. Fundamentals of Fluid Mechanics. 3rd ed. New York: John Wiley & Sons, 1998.
2. Anderson J D. Fundamentals of Aerodynamics. 2nd ed. New York: McGraw-Hill, 1991.
3. Streeter V L. E B Wylie and Bedford K W. Fluid Mechanics. 9th ed. New York: McGraw – Hill, 1998.
4. Fox R W and McDonald A. Introduction to Fluid Mechanics. 5th ed. New York: John Wiley & Sons, 1998.

第12章 热 传 导

以下给出分析和设计导热、对流和辐射传热的几种方法。

12.1 导热

12.1.1 具有表面对流的半无限大平板的瞬态导热

半无限大固体平板初始温度分布均匀，边界表面 $\eta=0$ 处具有对流，该平板瞬时温度分布由下式给出：[①]

$$\theta(\eta,\tau) = \mathrm{erfc}\left[\frac{\eta}{2\tau}\right] - \exp[\eta+\tau^2]\,\mathrm{erfc}\left[\frac{\eta}{2\tau}+\tau\right]$$

式中 erfc 是修正误差函数：

$$\theta(\eta,\tau)=\frac{T(\eta,\tau)-T_\infty}{T_\infty-T_i}$$

$$\tau=\frac{h}{k}\sqrt{\alpha t}$$

$$\eta=\frac{hx}{k}$$

x 是空间坐标，t 是时间，h 是传热系数，k 是固体的热导，T_∞ 是环境温度，α 是固体热扩散系数。

在 $0\leqslant\eta\leqslant5$ 和 $0.01\leqslant\tau\leqslant3$ 的范围内，绘出固体中温度分布曲线，然后把温度表示成 τ 的函数，其中 $0.01\leqslant\tau\leqslant4$，且 η 取 0,1,2,…,5 六个不同的值。程序清单如下：

```
tau = linspace(0.01,3,30);eta = linspace(0,5,20);
[x,t] = meshgrid(eta,tau);
theta = inline('erfc(0.5 * x./t) - exp(x + t.^2). * erfc(0.5 * x /t + t)','x','t');
figure(1)
mesh(x,t,theta(x,t))
xlabel('\eta')
ylabel('\tau')
zlabel('\theta')
figure(2)
eta = linspace(0,5,6);
tau = linspace(0 01,4,40);
for k = 1:6
  thet = theta(eta(k),tau);
  plot(tau,thet)
text(.92 * 4,1.02 * thet(end),['\eta = 'num2str(eta(k))]).
  hold on
end
```

---

① 参见 F.P Incropera 和 D.P DeWitt, *Fundamentals of Heat and Mass Transfer*，第四版，John Wiley & Sons, New York, 1996年，第239页。

```
xlabel('\tau')
ylabel('\theta')
```

执行程序后,得到图12.1和图12.2。

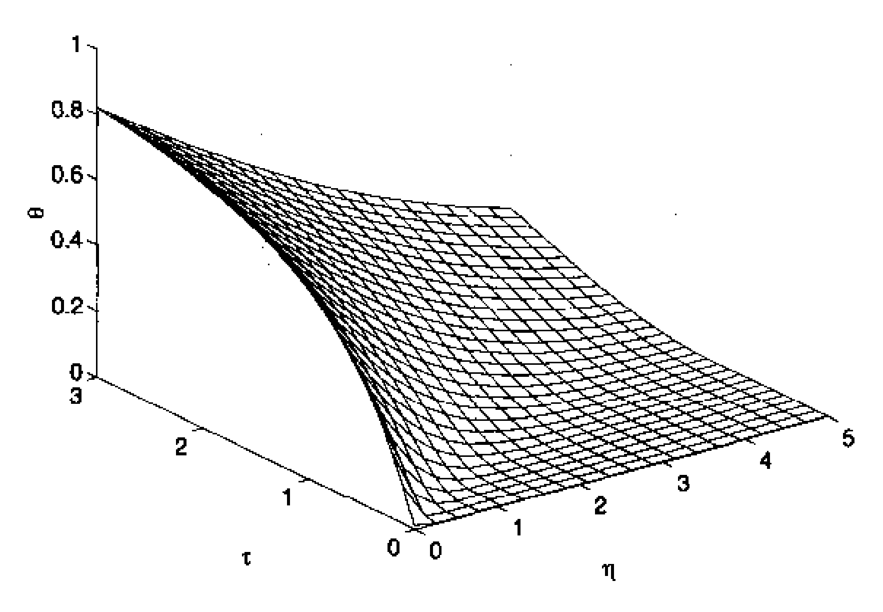


图12.1 半无限大固体温度随位置 η 和时间 τ 变化的格状曲面图

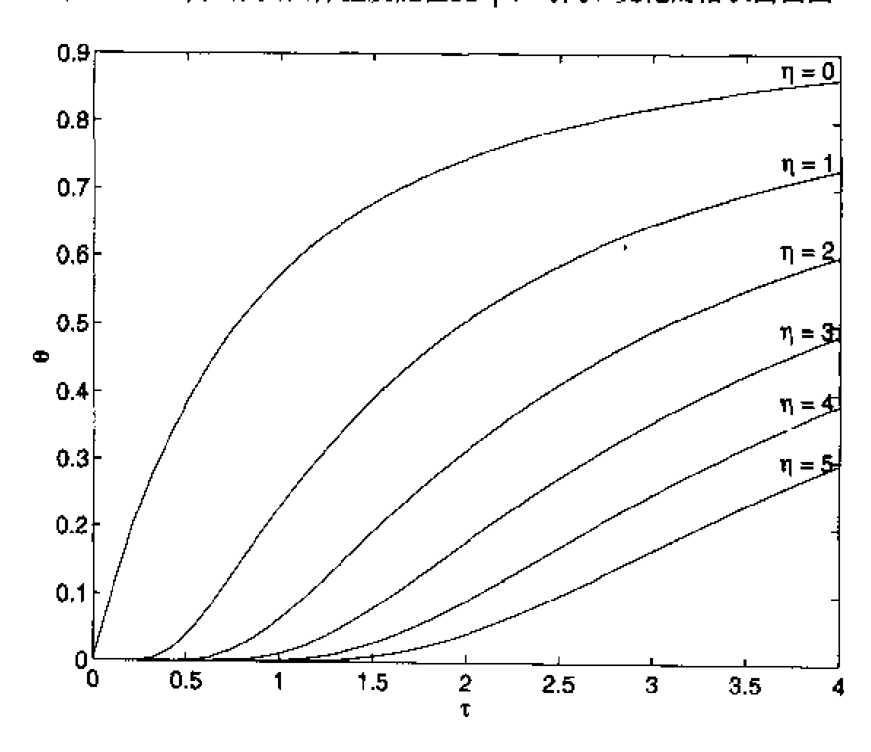


图12.2 半无限大固体温度随位置 η 和时间 τ 变化的曲线

12.1.2 具有对流的无限长圆柱体的瞬态导热

一个无限长圆柱固体初始温度均匀分布,表面具有对流,其瞬时温度分布由下式给出:①

① 参见F.P.Incropera和D.P.DeWitt,出处同上,第229页。

$$\theta(\xi,\tau) = \sum_{n=1}^{\infty} C_n \exp(-\zeta_n^2\tau) J_0(\zeta_n\zeta)$$

其中：

$$\theta(\xi,\tau) = \frac{T(\xi,\tau) - T_\infty}{T(\xi,0) - T_\infty}$$

而且，

$$C_n = \frac{2}{\zeta_n} \frac{J_1(\zeta_n)}{J_0^2(\zeta_n) + J_1^2(\zeta_n)}$$

$J_m(x)$是一类 m 阶贝塞尔函数，$\tau = \alpha t/a^2$，α 是热扩散系数，a 是圆柱体的半径，t 是时间，$\xi = r/a$，r 是圆柱体的径向距离，T_∞ 是环境温度，ζ_n 是下式的正根：

$$\frac{J_1(\zeta_n)}{J_0(\zeta_n)} - \frac{Bi}{\zeta_n} = 0$$

式中 $Bi = ha/k$ 是毕奥特数，h 是传热系数，k 是圆柱体的热导。

在区间 $0 \leqslant \xi \leqslant 1$ 和 $0 \leqslant \tau \leqslant 1.5$ 内且 $Bi = 0.5$ 时，用 ζ_n 的最小 15 个正根绘出 $\theta(\xi,\tau)$的分布图。程序如下：

```
Bi = .5;nroots = 15;r = zeros(1,nroots);
guess = 0.01;
CylinderRoots = inline('x. * besselj(1,x) - Bi * besselj(0,x)','x','Bi');
options = optimset('display','off');
for k = 1:nroots
  r(k) = fzero(CylinderRoots,[guess guess + 1.1 * pi],options,Bi);
  guess = 1.05 * r(k);
end
tau = linspace(0,1.5,20);
[t,rt] = meshgrid(tau,r);
Fn = exp( - t. * rt.^2);
cn = 2 * besselj(1,r)./(r. * (besselj(0,r).^2 + besselj(1,r).^2));
ccn = meshgrid(cn,tau);
pro = ccn'. * Fn;
rstar = linspace(0,1,20);
[R,rx] = meshgrid(rstar,r);
Jo = besselj(0,rx. * R);
the = Jo' * pro;
[rr,tt] = meshgrid(rstar,tau);
mesh(rr,tt,the')
xlabel(' \ xi')
ylabel(' \ tau')
zlabel(' \ theta')
view(49.5, - 34)
```

程序中的 *guess* 值由 CylinderRoots 函数的图形确定。程序结果见图 12.3。

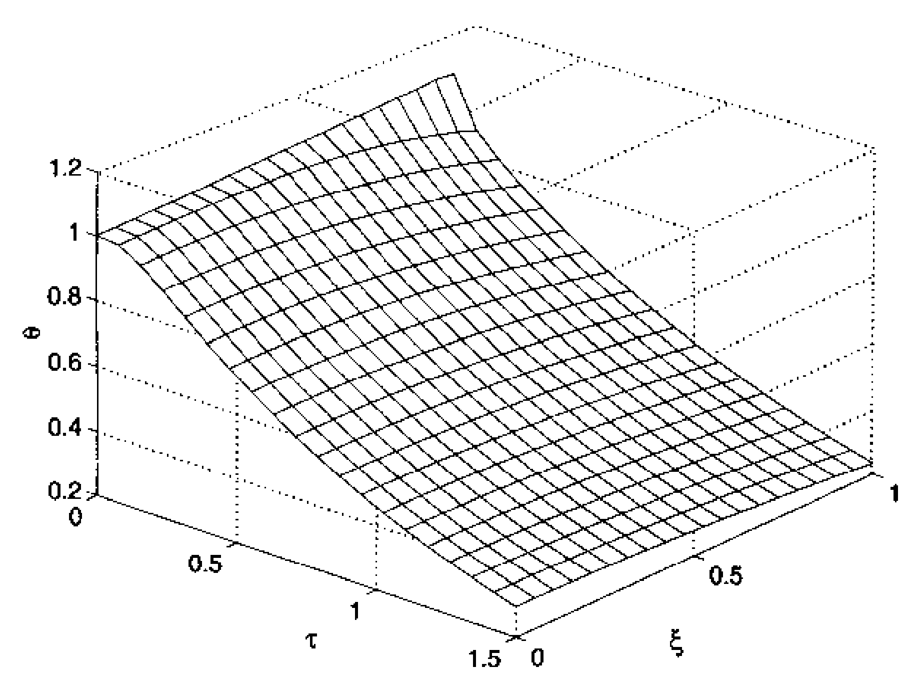


图 12.3 圆柱体内温度分布是时间 τ 和径向位置 ξ 的函数

12.1.3 有热源的一维瞬态导热

一维瞬态导热的特征方程为：

$$\frac{1}{\alpha}\frac{\mathrm{d}T}{\mathrm{d}t}=\frac{\mathrm{d}^2T}{\mathrm{d}x^2}+\frac{q}{k}$$

其中 T 是温度，t 是时间，x 是空间坐标，α 是热扩散系数，k 是热导系数，q 是体积热源。通过引入无量纲变量，把上式转化成无量纲形式：

$$\xi=\frac{x}{L}\quad \tau=\frac{\alpha t}{L^2}\quad Bi=\frac{hL}{k}$$

$$\theta=\frac{T-T_\infty}{T_i-T_\infty}\quad \sum=\frac{L^2q}{k(T_i-T_\infty)}\quad \chi=\frac{-q''L}{k(T_i-T_\infty)}$$

其中 L 是特征长度，T_i 代表初始温度，q''是热流量，T_∞ 为对流边界的流体温度。在非对流边界条件下，T_∞ 根据需要取值，但其值不能等于 T_i。特征方程利用这些变量变为：

$$\frac{\partial\theta}{\partial\tau}=\frac{\partial^2\theta}{\partial\xi^2}+\sum \tag{12.1}$$

区域两个边界上典型的边界条件有：

恒温：

$$\theta=\theta_w$$

恒定热流量：

$$\frac{\partial\theta}{\partial\xi}=\chi_w$$

对流：

$$\frac{\partial\theta}{\partial\xi}=-Bi\theta_w$$

式中负号是在对流条件下对左边界进行修正，下标 w 代表壁的值。

式(12.1)是一维抛物线偏微分方程。一种解法是变量分离法，适用于边界条件受限的情况。第12.1.2节的结果就是用这种方法得到的。偏微分方程(PDE)工具箱提供了另一种解法。由于它不适于只有一个空间变量的情况，所以解决这类问题比较麻烦。

这里利用 MATLAB 矩阵/向量解决空间的问题，给出在时间方向采用隐格式的有限差分法。图12.4给出了利用有限差分方法计算空间微元体的示意图。设有 N 个沿空间某一方向等距离分布的温度方格，每个格的温度需要计算出来。方格长为 $\Delta\xi = 1/(N-1)$，系统两端的方格长为$\Delta\xi/2$。在每个格上的差分方程为：

$$\frac{\mathrm{d}\theta_i}{\mathrm{d}\tau} = \frac{\theta_{i-1} - 2\theta_i + \theta_{i+1}}{\Delta\xi^2} + \Sigma \quad i = 2,\cdots,N-1$$

对一端是边界的网格有下列三种关系：

恒壁温：

$$\frac{\mathrm{d}\theta_i}{\mathrm{d}\tau} = 0 \quad i = 1 \text{ 或 } N$$

恒热流量：

$$\frac{\mathrm{d}\theta_i}{\mathrm{d}\tau} = \frac{2}{\Delta\xi^2}(\theta_I - \theta_i) + \Sigma - \frac{2\chi_w}{\Delta\xi} \quad i = 1 \text{ 或 } N$$

对流：

$$\frac{\mathrm{d}\theta_i}{\mathrm{d}\tau} = -\frac{2Bi\theta_i}{\Delta\xi} + \frac{2}{\Delta\xi^2}(\theta_I - \theta_i) + \Sigma \quad i = 1 \text{ 或 } N$$

其中 θ_I 是一个相邻的区间点：在左边界 $I = 2$，在右边界 $I = N-1$。

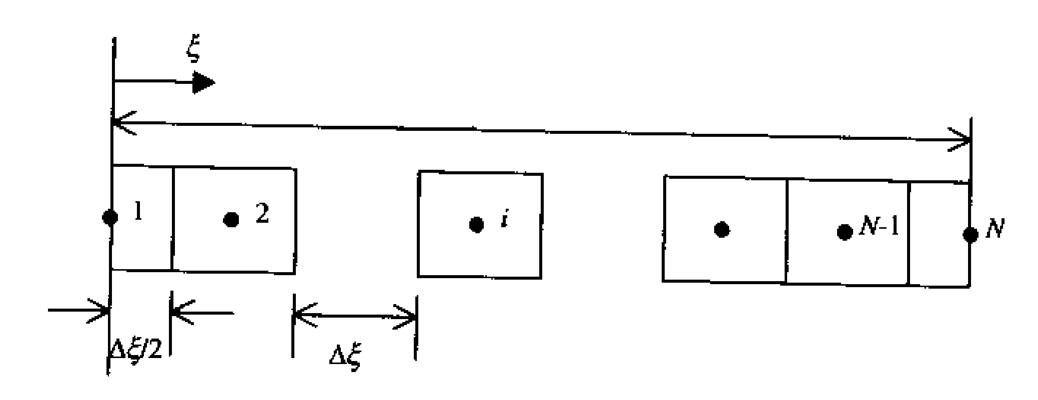


图12.4 有限差分原理图

表12.1 获得图12.5的边界条件和初始条件

参数	值
边界条件和源	
无量纲热源强度，Σ	1
左边界毕奥特数	0.1
右边界无量纲温度 $\theta(1)$	0.55
初始条件	
左边界 $\theta_i(0)$	1
$\theta_i(0) = 1$ 和 $\theta_i(1) = 0.55$ 间的线性分布	
有限差分参数	
网格个数	5
积分范围(归一化时间)	1

首先创建函数 FiniteDiffCond 计算有限差分方程中每一个格内的时间导数向量。在该函数中,左边界用对流边界条件,右边界用恒壁温条件。程序清单如下:

```
function dTdtau = FiniteDiffCond(t,T,flag,N,sigma,Bi)
dTdtau = zeros(N,1);
dx = 1/(N - 1);
dTdtau(1,1) = -2 * Bi * T(1)/dx + 2/dx^2 * (T(2) - T(1)) + sigma;
for i = 2:N - 1
  dTdtau(i,1) = ( -2 * T(i) + T(i-1) + T(i+1))/dx^2 + sigma;
end
dTdtau(N,1) = 0;
```

计算具有表 12.1 给出的边界条件和初始条件的系统,程序为:

```
D = 5;Sigma = 1;Bi = 0.1;
Tright = 0.55;Tleft = 1;tmax = 1;
Tp = linspace(Tleft,Tright,D)';
[t T] = ode15s('FiniteDiffCond',[0 1],Tp,[],D,Sigma,Bi);
plot(t,T,'k')
axis([0 1 0.5 1])
xlabel('\tau')
ylabel('\theta')
text(0.5,1.02 * T(end,5),'\xi = 1.0')
text(0.5,1.02 * T(end,4),'\xi = 0.75')
text(0.5,1.02 * T(end,3),'\xi = 0.5')
text(0.5,1.02 * T(end,1),'\xi = 0.0 and 0.25')
```

程序执行结果见图 12.5。该程序中用 ode15s 函数对无量纲时间 τ 取积分。相邻元素内温度的相互作用很强,这表明方程是一个刚性系统,利用 ode45 求解会导致不稳定。用函数 ode15s 来解决刚性系统情况可以得到平滑(稳定)的结果。

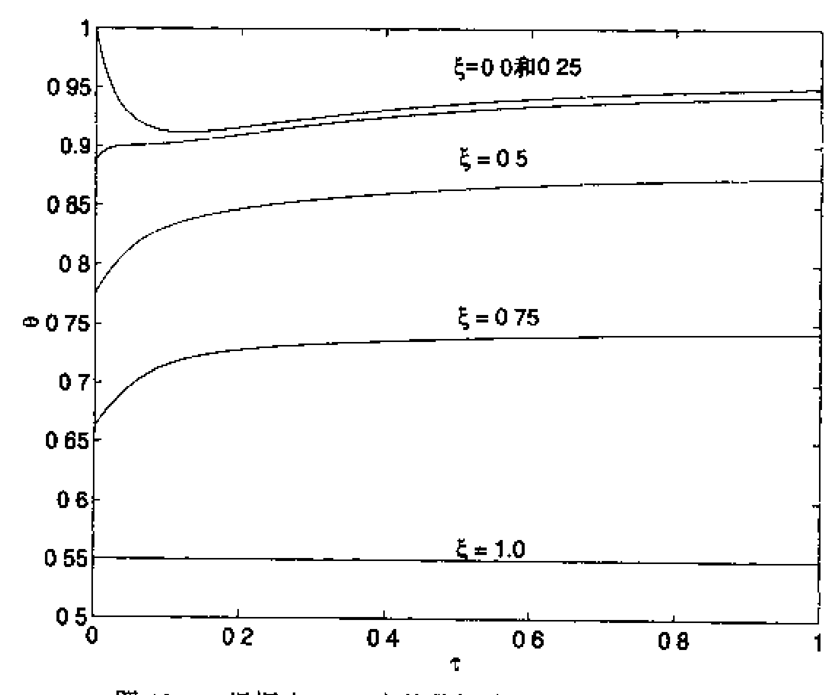


图 12.5 根据表 12.1 中的数据绘出的一维导热曲线图

12.2 壳管式热交换器的外形设计

根据表 12.2 和图 12.6 的定义，可以得出壳管式热交换器的特征方程。

表 12.2 与热交换器相关的符号定义

符号	单位	说明
c_p	J/(kgK)	恒压下的比热
d_o	m	管道外径
d_i	m	管道内径
f		管内流动摩擦系数
f_s		壳内流动摩擦系数
h_o	W/(m²K)	管内传热系数
h_i	W/(m²K)	管外传热系数
k	W/(m K)	流体热导函数
k_{tube}	W/(m K)	管道热导函数
$\dot{m}_t$	kg/s	管内质量流速
$\dot{m}_s$	kg/s	壳内质量流速
Δp_s	Pa	壳内压降
Δp_t	Pa	管内压降
u_t	m/s	管内流体的平均轴向速度($=\dot{m}_t/\rho A_t$)
A_o	m²	($=\pi d_o L N_T/N_P$)管外表面积单位流量
A_i	m²	($=\pi d_i L N_T/N_P$)管内表面积单位流量
A_s	m²	壳中心线或靠近中心线的流体横截面积
A_t	m²	全部横截面的单位流量($=\pi d_i^2 N_T/(4N_P)$)
B	m	障板间距
C	m	相邻两管的距离(见图 12.7)
C_L		管道的分布常数
C_{TP}		管道计算积累常数
D_e	m	壳等效直径
D_s	m	壳内径
F		逆流布置的 LMTD 修正系数
L	m	管长
N_b		隔板数[(L/B)取整]
N_T		管数
N_P		管口数
P_T	m	节距(见图 12.7)
P_P	W	管内流体的泵送功率
P_r		普朗特数
Q	W	传热率
R_{fo}	(m²K)/W	管外的污垢热阻
R_{fi}	(m²K)/W	管内的污垢热阻
Re_b		T_b 点的管内雷诺数
Re_s		T_b 点的壳内雷诺数
ΔT_m	℃,K	对数平均温差(LMTD, log mean temperature difference)
T_{h1}	℃,K	热流体的入口温度
T_{h2}	℃,K	热流体的出口温度
T_{c1}	℃,K	冷流体的入口温度
T_{c2}	℃,K	冷流体的出口温度
T_b	℃,K	整体温度
T_w	℃,K	壁温
U	W/(m²K)	A 口的整体平均传热系数
η_p		泵效率($0.80 \le \eta_p \le 0.85$)
ϕ_s		黏度修正系数
μ	kg/(s m)	动力学粘度
μ_b	kg/(s m)	T_b 点的动力学粘度
μ_w	kg/(s m)	T_w 点的动力学粘度
ρ	kg/m³	密度

热交换器的能量守恒方程为：

$$Q = UAF\Delta T_m = (\dot{m}c_p)_h(T_{h1} - T_{h2}) = (\dot{m}c_p)_c(T_{c2} - T_{c1})$$

其中下标 1 代表流入，下标 2 代表流出，下标 h 代表热流体，下标 c 代表冷流体。假设为逆流布置，对数平均温差如下：

$$\Delta T_m = \frac{\Delta T_1 - \Delta T_2}{\ln(\Delta T_1/\Delta T_2)}$$

$$\Delta T_1 = T_{h1} - T_{c2}$$

$$\Delta T_2 = T_{h2} - T_{c1}$$

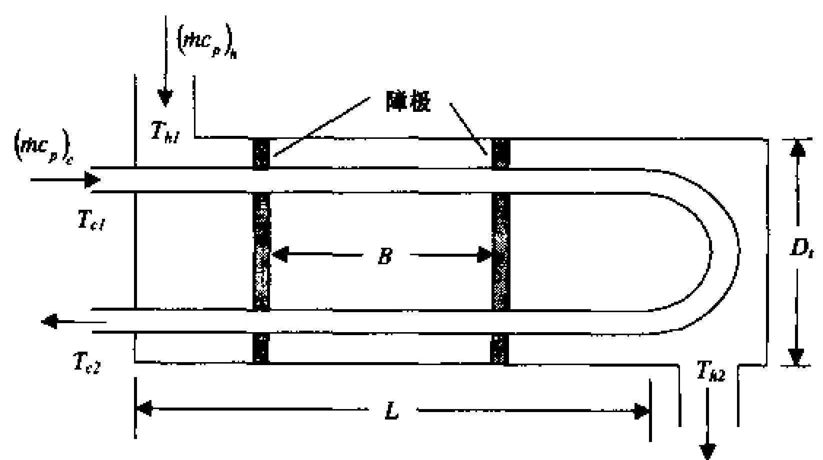


图 12.6 典型单壳程-双管程的壳管式热交换器

对于只有一个壳程，2 个、4 个等管程的壳管式热交换器①，有下式：

$$F = R_sC_0\left[\log_{10}\frac{C_1 + R_s}{C_1 - R_s}\right]^{-1}$$

对于有二个壳程，4 个、8 个等管程的壳管式热交换器，有下式：

$$F = 0.5R_sC_0\left[\log_{10}\frac{C_1 + C_2 + R_s}{C_1 + C_2 - R_s}\right]^{-1}$$

其中，

$$R_s = \sqrt{R^2 + 1} \quad P = \frac{T_{c2} - T_{c1}}{T_{h1} - T_{c1}} \quad R = \frac{T_{h1} - T_{h2}}{T_{c2} - T_{c1}}$$

$$C_1 = \frac{2}{P} - 1 - R \quad C_2 = \frac{2}{P}\sqrt{(1 - P)(1 - PR)}$$

$$C_o = \frac{1}{R - 1}\log_{10}\frac{1 - P}{1 - PR} \quad R \neq 1$$

$$= \frac{P}{2.3(1 - P)} \quad R = 1$$

总传热系数 U 由下式给出：

① 参见 R A Bowman，A C Mueller 和 W M Nagle，"Mean Temperature Difference in Design，" *Trans. ASME*，第 62 卷，1940 年，第 283 - 293 页。

$$U = \left[\frac{d_o}{d_i}\frac{1}{h_i} + \frac{d_o}{d_i}R_{fi} + \frac{d_o}{2k_{\text{tube}}}\ln\frac{d_o}{d_i} + R_{fo} + \frac{1}{h_o}\right]^{-1}$$

管内的传热系数值近似为：

$$h_i = \frac{k}{d_i}Nu_b$$

这里由于通过环形管道流体的性质不变，对于强制对流的充分发展的紊流流动有下式：[①]

$$Nu_b = \frac{0.125f(\text{Re}_b - 1000)\text{Pr}_b}{1 + 12.7\sqrt{0.125f}((\text{Pr}_b)^{2/3} - 1)} \quad 0.5 < \text{Pr}_b < 2000 \quad 3000 < \text{Re}_b < 5 \times 10^6$$

下标 b 表明该值是在管道内流体平均温度基础上计算出来的，称为流体平均温度。

f 是光滑管道内流动摩擦系数：

$$f = (0.790\ln\text{Re}_b - 1.64)^{-2} \quad 3000 \leqslant \text{Re}_b \leqslant 10^6$$

对于不光滑管道，令 $\lambda = f$，利用练习 5.5 的结果。

雷诺数由下式给出：

$$\text{Re}_b = \frac{\rho u_t d_i}{\mu} = \frac{4\dot{m}_t}{\pi d_i \mu N_T / N_p}$$

式中，

$$u_t = \dot{m}_t / (0.25\rho\pi d_i^2 N_T / N_p)$$

普朗特数为 Pr_b，由下式给出：

$$\text{Pr}_b = \frac{c_p\mu}{k}$$

壳内的传热系数可以用下式估计：

$$h_0 = 0.36\frac{k\phi_s}{D_e}(\text{Re}_s)^{0.55}(\text{Pr})^{1/3} \qquad 2000 < \text{Re}_s < 10^6$$

$$\phi_s = \left(\frac{\mu_b}{\mu_w}\right)^{0.14} \qquad T_w = (T_{c1} + T_{c2} + T_{h1} + T_{h2})/4$$

除了 ϕ_s 外，在壳内流体平均温度基础上算出其他值，如用温度 T_w 算出 μ_w。管道在壳内是成正方形铺设，如图 12.7 所示，壳的等效直径是 D_e，其表达式如下：

$$D_e = 4(P_T^2 - \pi d_0^2/4)/\pi d_0$$

对于管道成三角形铺设的情况：

$$D_e = 8(P_T^2\sqrt{3}/4 - \pi d_0^2/8)/\pi d_0$$

壳内雷诺数由下式给出：

$$\text{Re}_s = \frac{\dot{m}_s D_e}{A_s \mu}$$

$$A_s = 1.128CB\sqrt{\frac{N_T C_L}{C_{TP}}} = \frac{D_s CB}{P_T}$$

式中 D_s 是壳内径：

① 参见 F.P.Incropera 和 D.P.DeWitt，出处同上，第 424 页。

$$D_s = 1.128P_T\sqrt{\frac{N_T C_L}{C_{TP}}}$$

设 C_{TP} 的值分别为：

管程数为 1($N_p = 1$)：$C_{TP} = 0.93$
管程数为 2($N_p = 2$)：$C_{TP} = 0.90$
管程数为 3($N_p = 3$)：$C_{TP} = 0.85$

C_L 的值分别为：

$C_L = 1$ 对应于 90°和 45°
$C_L = 0.87$ 对应于 30°和 60°

给定壳内径(D_s)作为管道外径(d_o)的函数，标准管道铺设表可以给出所需管道数(N_T)、管道的节距(正方形或三角形)和管道数(N_p)。[①] 因此，数值计算结果通常接近标准值。

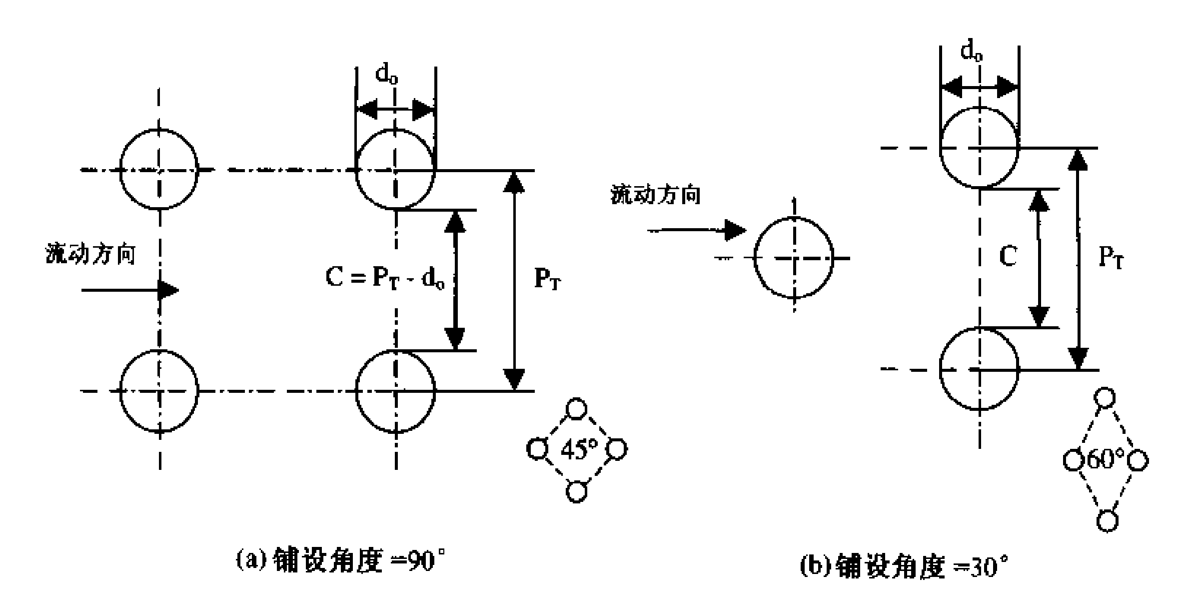


图 12.7 两种管道铺设

壳内的压降由下式给出：

$$\Delta p_s = \frac{f_s \dot{m}_s^2 (N_b + 1) D_s}{2A_s^2 \rho D_e \phi_s}$$

其中，

$$f_s = \exp(0.576 - 0.19\ln Re_s)$$

在温度为 T_b 时计算出这些值，管内的压降为：

$$\Delta p_t = f\left(\frac{LN_p}{d_i}\right)\left(\frac{\rho u_t^2}{2}\right)$$

忽略管道的弯曲因素，在温度 T_b 时计算出式中的值，仅适用于低速流动液体。

① 参见 S Kakac 和 H Liu, *Heat Exchangers: Selection, Rating and Thermal Design*, CRC Press, Boca Raton, FL, 1998 年，第 258～261 页。

泵送功率是：

$$P_p = \frac{\dot{m}_t \Delta p_t}{\rho \eta_p}$$

举例说明这些结果的应用。

例 12.1　确定管长和压降

用壳程为 1，管程为 2 的壳管式热交换器进行水水热交换，放热大约 800 kW。管道内是冷水，壳内是热水，初始分析结果见表 12.3。分析目标是：热交换器的长度是否可以小于 4.5 m，壳内和管内的压降是否可以小于 4 000 Pa。重新整理上面给出的方程，得出下式：

$$T_{h_2} = T_{h_1} - \frac{(\dot{m}c_p)_t}{(\dot{m}c_p)_s}(T_{c_2} - T_{c_1})$$

$$\Delta p_t = C_0 \frac{f\dot{m}_t^3}{UF\Delta T_m}$$

$$\Delta p_s = \frac{f_s \dot{m}_s^2 (N_b + 1) D_s}{2A_s^2 \rho D_e \phi_s}$$

$$L = \frac{(\dot{m}c_p)_t (T_{c_2} T_{c_1})}{\pi d_0 N_T UF\Delta T_m}$$

其中，

$$C_0 = \frac{N_p (T_{c_2} - T_{c_1})(c_p)_t}{2\pi d_i d_o N_T \rho_t A_t^2}$$

$$A_t = \frac{\pi d_i^2}{4} \frac{N_T}{N_P}$$

表 12.3　热交换器初步分析的参数

几何参数	物理参数
$D_s = 0.39$ m	$T_{c_1} = 18$℃
$N_T = 124$	$T_{c_2} = 42$℃
$N_P = 2$	
$P_T = 0.024$ m	$T_{h_1} = 65$℃
$B = 0.5$ m	
$N_b = 4$	$m_s = m_h = 14$ kg/s
$d_i = 16$ mm(0.016 m)	$m_t = m_c = 8.5$ kg/s
$d_o = 19$ mm(0.019 m)	$R_{fi} = 0\ 000 15$
$k_{tube} = 60$ W/m²K(碳钢)	$R_{fo} = 0\ 000 15$
90°管道铺设	
管道铺设成正方形	
光滑管道	

在计算之前，首先创建下列函数：

LMTDcorrFactor 用于计算 F

TubeFF 用于计算光滑管道的 f

WaterProperties 用于计算 0 到 100℃间在任意一温度点上 k，ρ，μ，c_p 和 Pr 的值

LMTD 用于计算对数平均温差

hTubeOutside 用于计算 h_o 和 Δp_s

hTubeInside 用于计算 h_i

PressureDropLength 用于计算 Δp_t 和 L

T2HotSide 调用 fzero 函数计算 T_{h_2}

WaterProperties 用表 12.4 给出的数据确定各种物理性能

函数 LMTDcorrFactor 的清单为：

```
function F = LMTDcorrFactor(Tc1,Tc2,Th1,Th2,NP)
P = (Tc2 - Tc1)/(Th1 - Tc1);
R = (Th1 - Th2)/(Tc2 - Tc1);
C1 = 2/P - 1 - R
C2 = 2 * sqrt((1 - P) * (1 - P * R))/P;
Rs = sqrt(R^2 + 1);
if R = = 1
  Co = R/2.3/(1 - P);
else
  Co = log10((1 - P)/(1 - P * R))/(R - 1);
end
if NP = = 1
  F = Rs * Co/log10((C1 + Rs)/(C1 - Rs));
else
  F = 0.5 * Rs * Co/log10((C1 + C2 + Rs)/(C1 + C2 - Rs));
end
```

函数 TubeFF 的清单为：

```
function f = TubeFF(Re)
f = 1/(0.79 * log(Re) - 1.64^2;
```

函数 WaterProperties 的清单为：

```
function[cp,mu,k,rho,Pr] = WaterProperties(Temp)
Temp = Temp + 273;
T = [273 285 300 315 330 345 360 373];
cpp = [4217 4189 4179 4179 4184 4191 4203 4217];
Muu = [1750 1225 855 631 489 389 324 279] * 1e - 6;
kk = [569 590 613 634 650 668 674 680] * 0.001;
rhoo = [1000 1000 998 991.1 1984.3 976.6 967.1 957.9];
Prr = [12.99 8.81 5.83 4.16 3.15 2.45 2.02 1.76];
cp = spline(T,cpp,Temp);
mu = spline(T,muu,Temp);
k = spline(T,kk,Temp);
rho = spline(T,rhoo,Temp);
Pr = spline(T,Prr,Temp);
```

函数 LMTD 的清单为：

```
function Tm = LMTD(Tc1,Tc2,Th1,Th2)
DT1 = Th1 - Tc2;
DT2 = Th2 - Tc1;
Tm = (DT1 - DT2)/log(DT1/DT2);
```

函数 hTubeInside 的清单为：

```
function hi = hTubeInside(Reb,Prb,kb,di)
f = TubeFF(Reb);
Nub = 0.125 * f * (Reb - 1000) * Prb/(1 + 12.7 * sqrt(0.125 * f) * (Prb^(2/3) - 1));
hi = Nub * kb/di;
```

函数 hTubeOutside 的清单为：

```
function[ho,DeltaPs] = hTubeOutside(Tb,Tw,ms,Ds,C,B,PT,do,pitch,Nb)
[cpb,mub,kb,rhob,Prb] = WaterProperties(Tb);
[cpw,muw,kw,rhow,Prw] = WaterProperties(Tw);
phis = (mub/muw)^0.14;
if pitch = = 'spuare'
  De = 4 * (PT^2 - pi * do^2/4)/pi/do;
else
  De = 8 * (PT^(sqrt(3)/4) - pi * do^2/8)/pi/do;
end
As = Ds * C * B/PT;
Res = ms * De/mub/As;
ho = 0.36 * kb * phis * Pes^0.55 * Prb^(1/3)/De;
fs = exp(0.576 - 0.19 * log(Res));
DeltaPs = fs * ms^2 * (Nb + 1) * Ds/(2 * As^2 * rhob * De * phis);
```

函数 T2HotSide 的清单为：

```
function Th = T2HotSide(Th2,mt,ms,Tc1,Tc2,Th1)
cph = WaterProperties((Th2 + Th1)/2);
cpc = WaterProperties((Tc1 + Tc2)/2);
Th = Th1 - mt * cpc * (Tc2 - Tc1)/ms/cph - Th2;
```

函数 PressureDropLength 的清单为：

```
function [DeltaPt,DeltaPs,L] = PressureDropLength(mt,Tc1,Tc2,Th1,Th2,ms,...
                                    di,do,NT,NP,Ds,C,B,PT,pitch,kTube,Rfi,Rfo)
Tcb = (Tc1 + Tc2)/2;
[cpc,muc,kc,rhoc,Prc] = WaterProperties(Tcb);
[cph,muh,kh,rhoh,Prh] = WaterProperties((Th1 + Th2)/2;
Th2 = Th1 - mt * cpc * (Tc2 - Tc1)/ms/cph;
Thb = (Th1 + Th2)/2;
At = 0.25 * pi * di^2 * NT/NP;
Co = 2 * NP * (Tc2 - Tc1) * cpc/pi/di/do/NT/rhoc/At^2;
Tm = LMTD(Tc1,Tc2,Th1,Th2);
F = LMTDcorrFactor(Tc1,Tc2,Th1,Th2,NP);
Tw = (Th1 + Th2 + Tc1 + Tc2)/4;
[ho,DeltaPs] = hTubOutside(Thb,Tw,ms,Ds,C,B,PT,do,pitch,Nb);
Rec = 4 * mt * NP/pi/di/muc/NT;
hi = hTubeInside(Rec,Prc,kc,di);
U = 1/(do/di/hi + do * Rfi/di + do/2/kTube * log(do/di) + Rfo + 1/ho);
f = TubeFF(Rec);
DeltaPt = Co * f * mt^3/U/F/Tm;
Q = mt * cpc * (Tc2 - Tc1);
L = Q/pi/do/NT/U/F/Tm;
```

程序为：

```
Ds = 0 39;NT = 124;NP = 2;B = 0 5;PT = 0 024;
di = 0 016;do = 0 019;kTube = 60;
Rfi = 0 00015;Rfo = 0 00015;
pitch = 'square';C = PT - do;ms = 14;mt = 8.5;
Tc1 = 18;Tc2 = 42;Th1 = 65;Nb = 4;
options = optimset('display','off');
Th2 = fzero('T2HotSide',Th1 + 15,options,mt,ms,Tc1,Tc2,Th1);
[DeltPt,DeltaPs,L] = PressureDropLength(mt,Tc1,Tc2,Th1,Th2,ms,...
                              di,do,NT,NP,Ds,C,B,PT,pitch,kTube,Rfi,Rfo,Nb);
disp(['Shell side exit temperature = 'num2str(Th2)'deg C'])
disp(['Tube side pressure drop = 'num2str(DeltaPt)'Pa'])
disp(['Shell side pressure drop = 'num2str(DeltaPs)'Pa'])
disp(['Tube length = 'num2str(L)'m'])
```

程序执行后的结果在 MATLAB 命令窗口中显示：

```
Shell side exit temperature = 50.4487 deg C
Tube side pressure drop = 3568.8913 Pa
Shell side pressure drop = 1799.4813 Pa
Tube length = 4.232 m
```

可以看出该结果满足设计要求。

表 12.4 水的热物理性能

$T(K)$	k(W/mK)	ρ(kg/m³)	μ(Ns/m²)	c_p(J/kgK)	Pr
273	0.569	1 000.0	$1\,750\times10^{-6}$	4 217	12.99
285	0.590	1 000 0	$1\,225\times10^{-6}$	4 189	8.81
300	0.613	998 0	855×10^{-6}	4 179	5.83
315	0 634	991 1	631×10^{-6}	4 179	4.16
330	0 650	984 3	489×10^{-6}	4 184	3.15
345	0 668	976 6	389×10^{-6}	4 191	2.45
360	0 674	967 1	324×10^{-6}	4 203	2.02
373	0 680	957.9	279×10^{-6}	4 217	1.76

12.3 对流传热

12.3.1 平板热边界层——相似解

图 12.8 给出了流过平板的层流边界层的速度纵剖图，该图通过求解下面的勃拉修斯方程获得。[①]

$$\frac{d^3f}{d\eta^3}+\frac{f}{2}\frac{d^2f}{d\eta^2}=0$$

① 参见 F P Incropera 和 D P DeWitt，出处同上，第 350～352 页。

其中 f 是一个修正的流函数：

$$f = \frac{\Psi}{u_\infty \sqrt{v_{ms} x / u_\infty}}$$

流函数 Ψ 定义为：

$$u = \partial \Psi / \partial y$$
$$v = -\partial \Psi / \partial x$$

其中 u 和 v 分别是对应 x、y 方向上的速度，η 是相似变量：

$$\eta = y \sqrt{u_\infty / v_{ms} x}$$

来流速度为 u_∞，流体的运动粘度是 v_{ms}。通过求解勃拉修斯方程可以得到边界层上任意一点的速度。第 11.3.2 节给出了勃拉修斯方程的数值解。

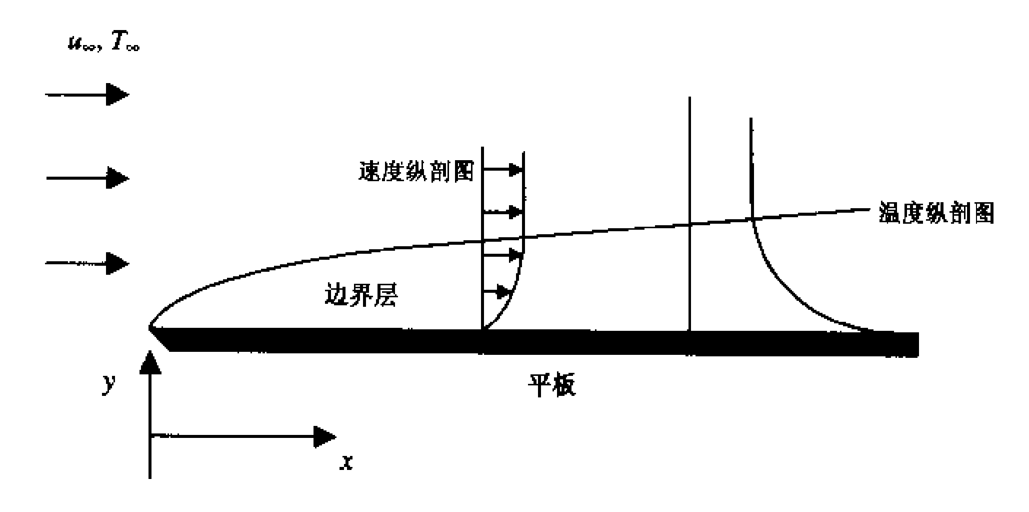


图 12.8　平板上的流动

在稳流的情况下，对边界层进行一些假设，那么流体的能量方程可以通过相似变量获得。[①]

$$\frac{d^2 T^*}{d\eta^2} + \Pr \frac{f}{2} \frac{dT^*}{d\eta} = 0$$

其中 T^* 是无量纲的温度值：

$$T^* = \frac{T - T_s}{T_\infty - T_s}$$

T 是流体温度，T_s 是平板表面温度，T_∞ 是流体来流的温度，普朗特数是 $Pr = V_{ms}/\alpha$，其中 α 是流体的热扩散率。注意：通过能量方程中的 f, T^* 和流速相耦合。

边界条件为：

$$f(0) = 0 \quad \left.\frac{df}{d\eta}\right|_{\eta=0} = 0 \quad \left.\frac{df}{d\eta}\right|_{\eta=\infty} \to 1$$

$$\left.\frac{dT^*}{d\eta}\right|_{\eta=0} = 0 \quad T^*(\eta \to \infty) \to 1$$

求解扩展的勃拉修斯方程方法与第 11.3.2 节求解勃拉修斯方程的方法相似。在这个例

① 参见 F P Incropera 和 D P DeWitt，出处同上。

子中,函数 fsolve 用来确定在 $\eta=0$ 时未知的边界条件。当 $\eta\to\infty$ 时边界条件近似满足。通过引入以下变量可以把这两个相关的非线性方程转换为五个相关的一阶常微分方程:

$$y_1=f \qquad y_4=T^*$$

$$y_2=\frac{df}{d\eta} \qquad y_5=\frac{dT^*}{d\eta}$$

$$y_3=\frac{d^2f}{d\eta^2}$$

其中 y_1 为流函数,y_2 为速度,y_3 是交量,y_4 为温度,y_5 为热流量。

这些量由五个一阶微分方程决定:

$$\frac{dy_1}{d\eta}=y_2 \qquad \frac{dy_4}{d\eta}=y_5$$

$$\frac{dy_2}{d\eta}=y_3 \qquad \frac{dy_5}{d\eta}=-\frac{\Pr}{2}y_1y_5$$

$$\frac{dy_3}{d\eta}=-\frac{1}{2}y_1y_3$$

相应的边界条件为:

$$y_1(0)=0 \qquad y_4(\eta\to\infty)\to 1$$

$$y_2(0)=0 \qquad y_5(0)=0$$

$$y_2(\eta\to\infty)\to 1$$

函数 ode45 要求在 η 的某个值处所有的边界条件必须确定,在本例中 $\eta=0$。$\eta\to\infty$ 时的两个边界条件必须由 $\eta=0$ 时的相应边界条件代替。它们是:

$$y_3(0)=a$$

$$y_4(0)=b$$

通过调用 fsolve 函数来获得 a 和 b 的值,当 $\eta\to\infty$ 时,边界条件得到满足。

在第 11.3.2 节中介绍的 Blasius(勃拉修斯)函数可以用来求解平板上的流速。同样,也可以调用该函数求解扩展方程,其中包括求解能量方程。由于前面已对该函数进行了描述,所以这里就不再重复 。

```
function F = Blasius(x,y)
F = [y(2);y(3);-0.5*y(1)*y(3)];
```

现在创建函数 BlasiusT 来计算 yi,该函数由 ode45 函数在积分时调用,作为计算过程的一个部分,该函数调用 Blasius。

```
function F = BlasiusT(x,y,flag,Pr)
F = [Blasius(x,y)(1:3));y(5);-Pr*0.5*y(1)*y(5)];
```

最后,创建一个 BlasiusT2 函数,通过调用 ode45 函数对五个微分方程进行积分,该函数的输入包括普朗特数 Pr 和一个二维向量 $\boldsymbol{x}$。$\boldsymbol{x}$ 是未知边界条件 a 和 b 的值。通过积分获得一个足够大的接近于 η_{max} 的值 η,η_{max} 的值依赖于普朗特数以及所要求的精度。一般说来,当 Pr 的值较小时,要求 η_{max} 的值较大。BlasiusT2 函数的输出结果是一个二维向量,分别代表 $1-y_2(\eta_{max})$ 和 $1-y_4(\eta_{max})$ 的值,经过迭代运算 fsolve 函数使向量趋近于 0。

```
function fn = BlasiusT2(x,Pr,eta_max)
xspan = [0 eta_max];
y0 = [0 0 x(1)0 x(2)];
[eta ff] = ode45('BlasiusT',xspan,y0,[],Pr);
fn = [1 - ff(end,2),1 - ff(end,4)];
```

程序执行后的结果在下面给出。对应 Pr = 0.07、0.7 和 7.0 时的结果分别见图 12.9 至图 12.11。η_{max} = 8 对应 Pr = 0.7 和 7.0；η_{max} = 20 对应 Pr = 0.07。

```
Pr = [0.07 .7 7];etaMax = [1 5 8 8];xm = [1 5 5 5];
x0 = [.3 .3];
options = optimset('display','off');
for k = 1:3
 figure(k)
 slopeW = fsolve('BlasiusT2',x0,options,Pr(k),etaMax(k));
 y0 = [0 0 slopeW(1) 0 slopeW(2)];
 xspan = [0 etaMax(k)];
 [eta ff] = ode45('Blasius T',xspan,y0,[],Pr(k));
 subplot(2,1,1)
 plot(eta,ff(:,1),':k',eta,ff(:2),'-k',eta,ff(:,3),'--k')
 legend('Stream function','Velocity','Shear')
 axis([0 xm(k) 0 2])
 xlabel('\eta')
 ylabel('y_i(i=1,2,3)')
title(['Pr='num2str(Pr(k))])
 subplot(2,1,2)
 plot(eta,ff(:,4),'-',eta,ff(:,5),'--')
 axis([0 xm(k)0 2])
 legend('Temperature','Heat flux')
 xlabel('\eta')
 ylabel('y_i(i=4,5)')
end
```

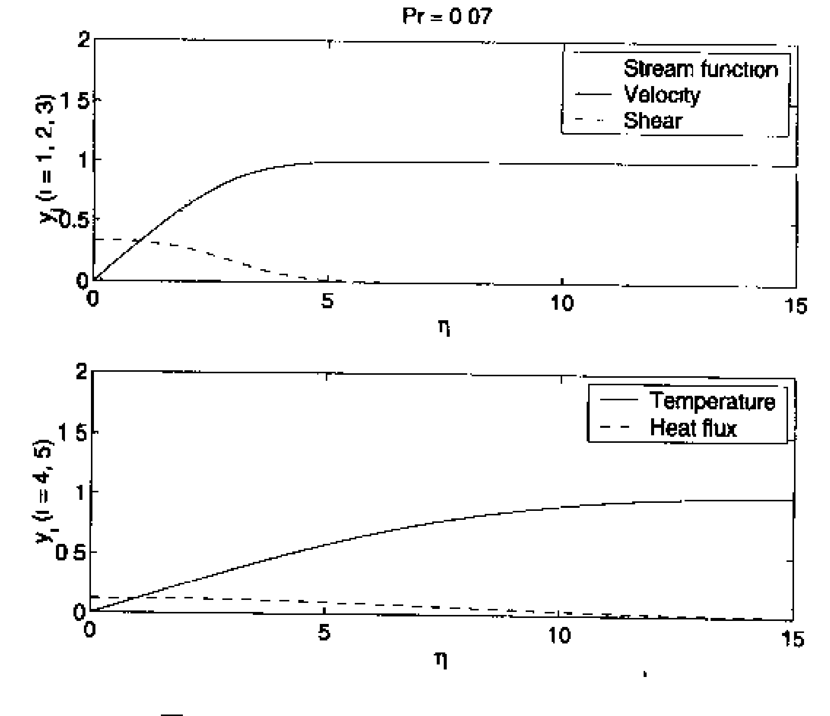


图 12.9 Pr = 0.07 时扩展勃拉休斯方程的解

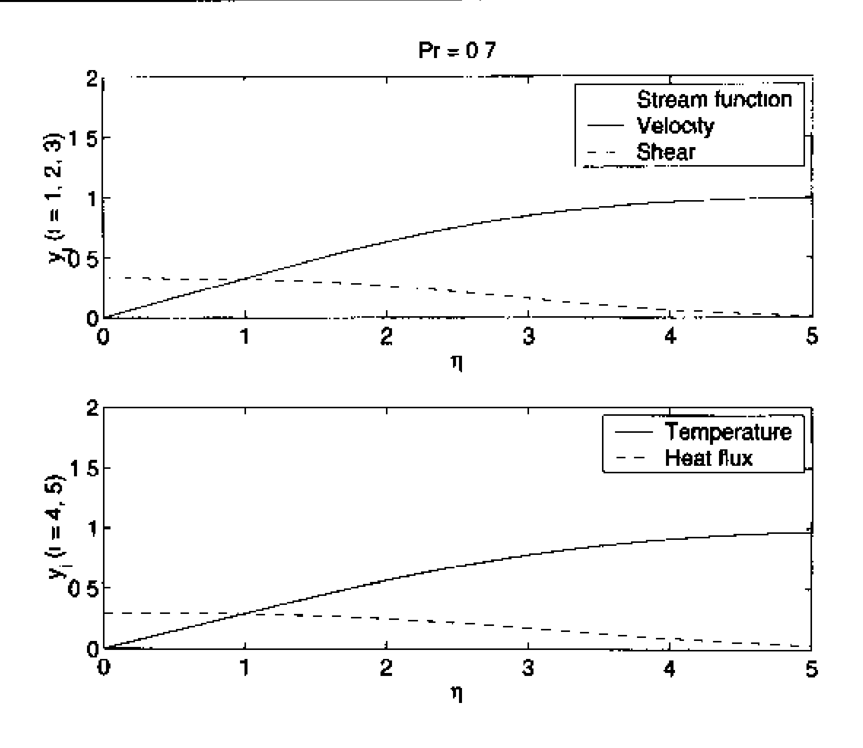


图 12.10 Pr=0.7时扩展勃拉休斯方程的解

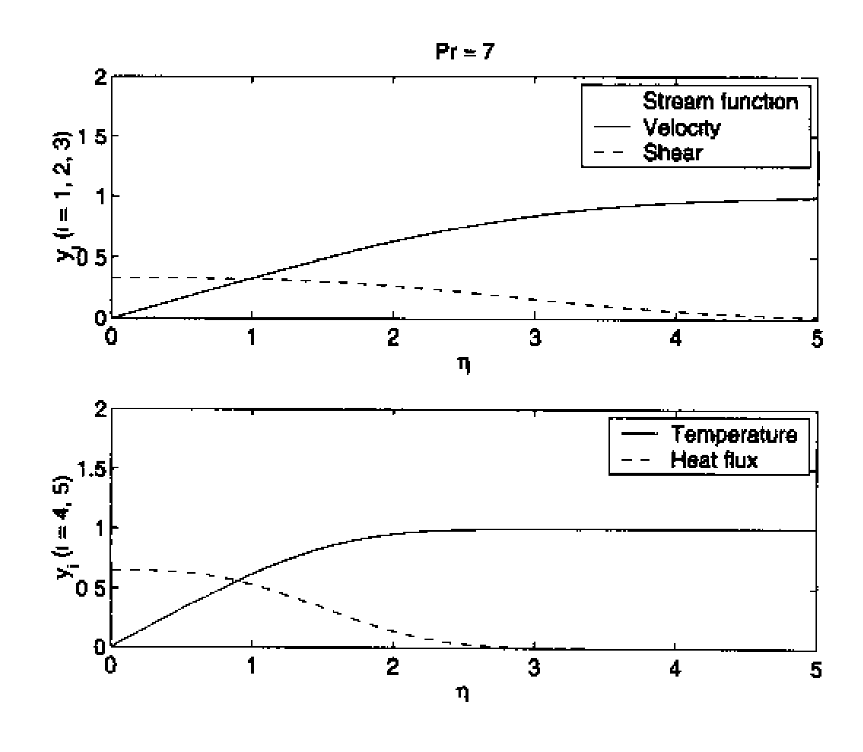


图 12.11 Pr=7.0时扩展勃拉休斯方程的解

12.3.2 自然对流相似解

竖直的加热板与冷的液体相接触,它们之间的自然对流如图 12.12 所示。大部分的流体是静止的,但是由于平板上的热交换造成浮升力流动,这种流动可以由下面给出的两个相关的

非线性常微分方程描述:[①]

$$\frac{d^3 f}{d\eta^3} + 3f\frac{d^2 f}{d\eta^2} - 2\left(\frac{df}{d\eta}\right)^2 + T^* = 0$$

$$\frac{d^{2*} T^*}{d\eta^2} + 3\Pr f\frac{dT^*}{d\eta} = 0$$

式中 f 是修正流函数:

$$f = \frac{\Psi}{4v_{\infty}(Gr_x/4)^{0.25}}$$

流函数 Ψ 定义为:

$$u = \partial\Psi/\partial y$$

$$v = -\partial\Psi/\partial x$$

式中 u 和 v 分别为沿 x 和 y 方向上的速度。η 为相似变量:

$$\eta = \frac{y}{x}\left(\frac{Gr_x}{4}\right)^{0.25}$$

格拉晓夫(Grashof)数定义如下:

$$Gr_x = g\beta(T_s - T_\infty)x^3/v_{\infty}^2$$

式中 g 是重力加速度,β 是热膨胀系数,为:

$$\beta = \frac{-1}{\rho}\left(\frac{\partial\rho}{\partial T}\right)_p$$

v_{∞}是运动粘度,Pr 是先前定义的普朗特数,T^* 是无量纲温度,表达式如下:

$$T^* = \frac{T - T_\infty}{T_s - T_\infty}$$

该系统的边界条件为:

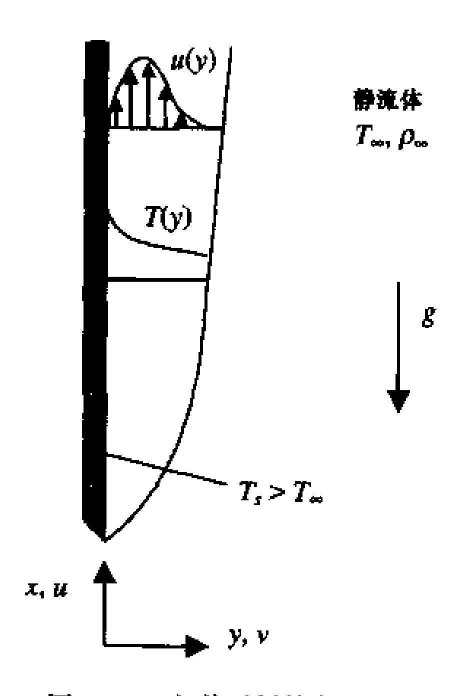


图 12.12 加热平板的自然对流

① 参见 F.P.Incropera 和 D.P.DeWitt,出处同上,第 487~490 页。

$$\eta = 0: f = 0 \quad \frac{df}{d\eta} = 0 \text{和} T^* = 1$$

$$\eta \to \infty: \quad \frac{df}{d\eta} \to 0 \text{和} T^* \to 0$$

通过引入下面的中间变量,可以将前面的系统用五个一阶微分方程表示:

$$y_1 = f \qquad y_4 = T^*$$

$$y_2 = \frac{df}{d\eta} \qquad y_5 = \frac{dT^*}{d\eta}$$

$$y_3 = \frac{d^2 f}{d\eta^2}$$

含有新变量的微分方程如下:

$$\frac{dy_1}{d\eta} = y_2 \qquad \frac{dy_4}{d\eta} = y_5$$

$$\frac{dy_2}{d\eta} = y_3 \qquad \frac{dy_5}{d\eta} = -3\Pr y_1 y_5$$

$$\frac{dy_3}{d\eta} = 2y_2^2 - 3y_1 y_3 - y_4$$

相应的边界条件为:

$$y_1(0) = 0 \qquad y_4(0) = 1$$

$$y_2(0) = 0 \qquad y_4(\eta \to \infty) \to 0$$

$$y_2(\eta \to \infty) \to 0$$

求解这些方程的麻烦之处在于,边界条件没有全部定义在 $\eta = 0$ 处,因此可以采用与求解勃拉修斯方程相似的方法。通过下式可以得到 a, b 的值:

$$y_3(0) = \left.\frac{d^2 f}{d\eta^2}\right|_{\eta=0} = a \qquad y_5(0) = \left.\frac{dT^*}{d\eta}\right|_{\eta=0} = b$$

因此,

$$\eta \to \infty: y_2 = \frac{df}{d\eta} \to 0 \text{和} y_4 = T^* \to 0$$

创建函数 NaturalConv,计算 5 个中间变量 y_i,然后调用函数 ode45,得到结果:

```
function ff = NaturalConv(x,y,flag,Pr)
ff = [y(2);y(3); -3 * y(1) * y(3) + 2 * y(2)^2 - y(4);y(5); -3 * Pr * y(1) * y(5)];
```

再创建函数 NaturalConv2,它调用函数 ode45 对五个一阶常微分方程积分。函数输入包括普朗特数 Pr 和一个二维向量 $\boldsymbol{x}$,该向量代表初始边界条件未给定时 a 和 b 试探值。y 的积分上限 η_{max} 取决于普朗特数,当普朗特数较大时,η_{max} 相应减小。随着 η_{max} 增大,结果与 η_{max} 的取值无关,即得到精确值的解。该函数的输出结果是一个二维向量,分别为 $y_2(\eta_{max})$ 和 $y_4(\eta_{max})$ 的值。在迭代过程中重复调用 fsolve 函数可使向量趋于零。程序如下:

```
function fn = NaturalConv2(x,Pr,etaMax)
xspan = [0 etaMax];
y0 = [0 0 x(1) 1 x(2)];
[eta ff] = ode45('NaturalConv',xspan,y0,[],Pr);
fn = [ff(end,2),ff(end,4)];
```

当 Pr 分别为 0.07、0.7 和 7.0 时，执行下面的程序得到如图 12.13 至图 12.15 所示的解，该系统的结果对 a 和 b 的初值很敏感。程序如下：

```
Pr = [0.07 0 7 7]; etaMax = [20 20 5]; xm = [1 5 5 5];
x01 = [0 9 0.7 0.5]; x02 = [ -0 2 -0.5 -1]; ym = [2 0.8 0.5];
options = optimset('display','off');
for k = 1:3
  x0 = [x01(k) x02(k)];
  slopeW = fsolve('NaturalConv2',x0,options,Pr(k),etaMax(k));
  y0 = [0 0 slopeW(1) 1 slopeW(2)]
  xspan = [0 etaMax(k)];
  [eta ff] = ode45('NaturalConv',xspan,y0,[],Pr(k));
  figure(k)
  subplot(2,1,1)
  plot(eta,ff(:,1),':k',eta,ff(:,2),'-k',eta,ff(:,3),'--k')
  legend('Stream function','Velocity','Shear')
  axis([0 xm(k) -0.2 ym(k)])
  xlabel('\eta')
  ylabel('y_i(i=1,2,3)')
  title(['Pr='num2str(Pr(k))])
  subplot(2,1,2)
  plot(eta,ff(:,4),'-',eta,ff(:,5),'--')
  axis([0 xm(k) -1.2 1])
  legend('Temperature','Heat flux')
  xlabel('\eta')
  ylabel('y_i(i=4,5)')
end
```

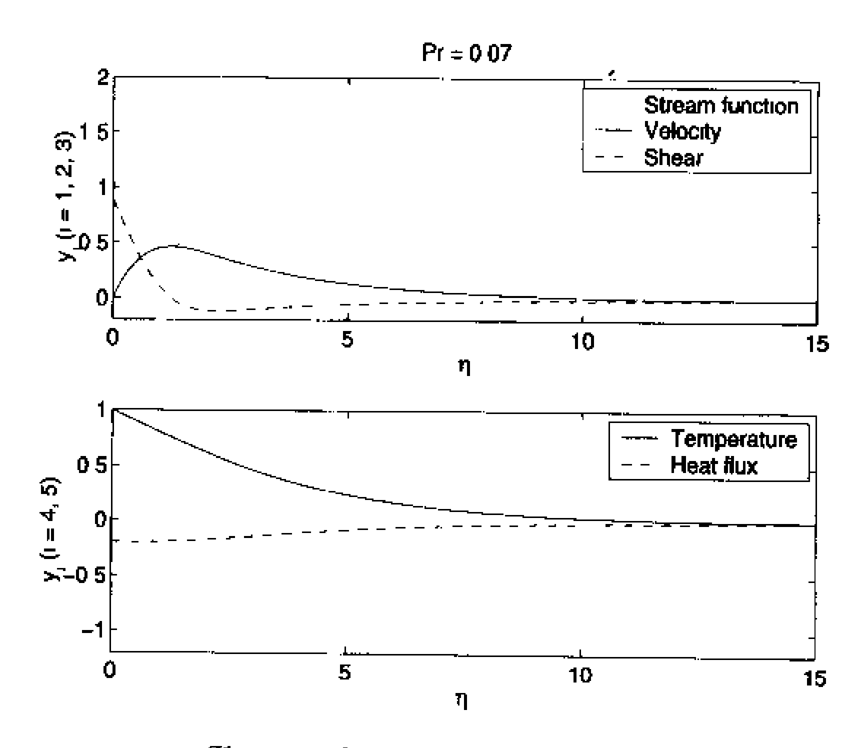


图 12 13 当 Pr = 0 07 时的自然对流的解

因为对于某些普朗特数通常认为积分限 η 无限大，所以对结果必须进行实验以确定 η 值。在这个例子中，对于 Pr = 0.07 和 0.7，积分限为 $0 < \eta < \eta_{max} = 20$；对于 Pr = 7.0，积分限为 $0 < \eta <$

$\eta_{max}=5$。按这种方法可确定出表12.5给出的壁上未知变量的值，其他普朗特数可照此法计算。

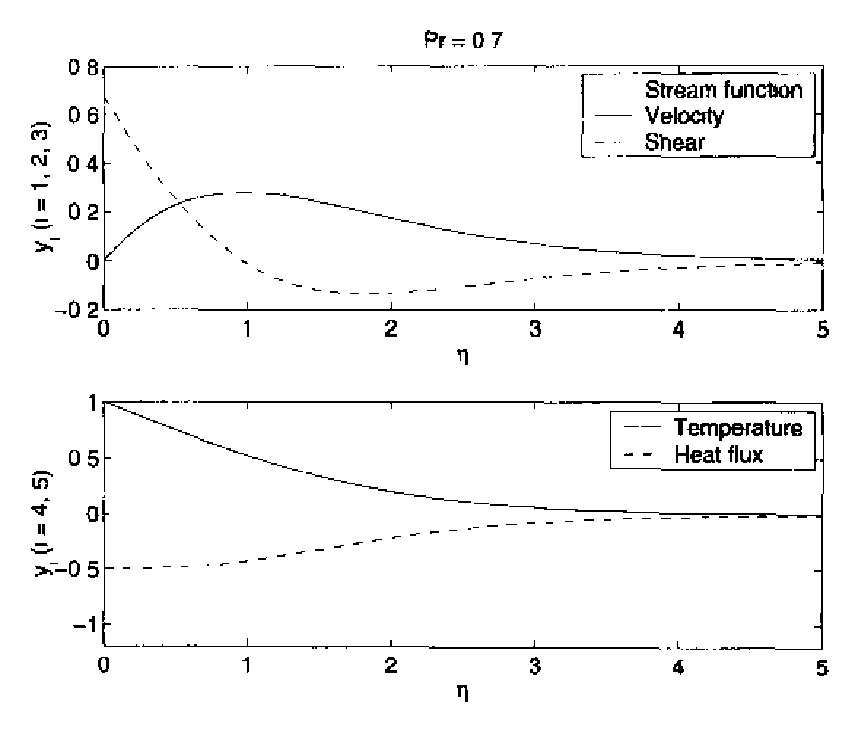


图 12.14 当 Pr = 0.7 时的自然对流的解

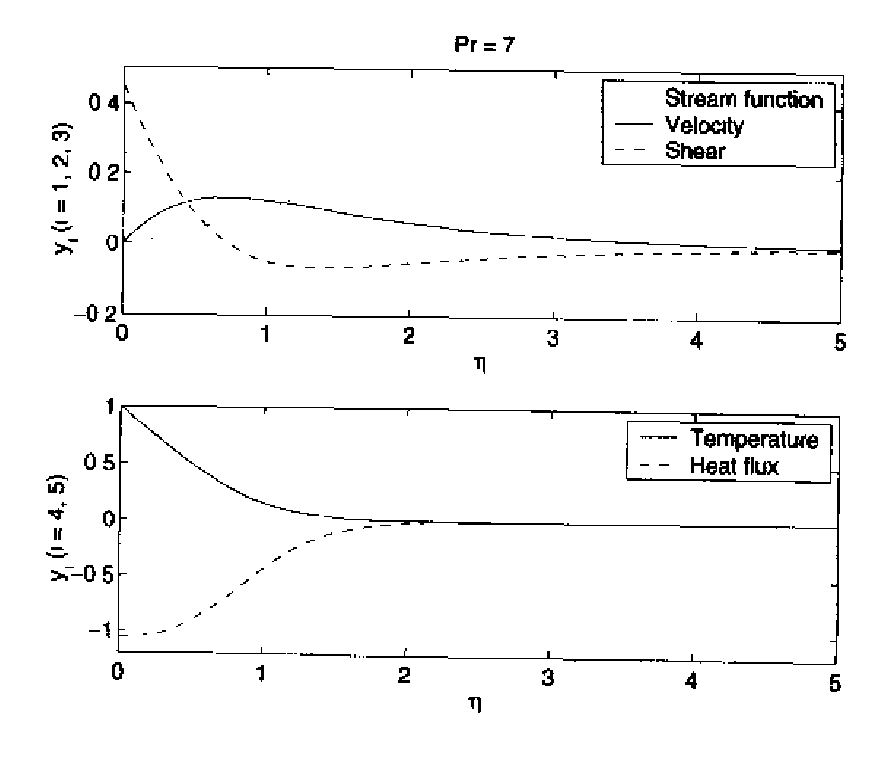


图 12.15 当 Pr = 7 0 时的自然对流的解

参见图12.13至图12.15，分别对应的普朗特数为0.07、0.7和7.0。可以看出，对以上三种情况，η在1附近时速度达到最大值，流体中平行于壁的剪应力为：

$$\tau_s = \frac{\sqrt{2}v^2\rho}{x^2}Gr_x^{3/4}\frac{d^2 f}{d\eta^2}$$

在速度达到最大值时上式趋于0。

表 12.5 $y_3(0)$和 $y_5(0)$的计算结果

Pr	$y_3(0)=\mathrm{d}f^2/\mathrm{d}\eta^2=a$	$y_t(0)=\mathrm{d}T^*/\mathrm{d}\eta=b$
0 07	0 885	-0 197
0 7	0 68	-0 50
7 0	0 45	-1 05

热导致流体流动,其结果是温度和速度边界层的厚度近似并与普朗特数无关。这一点与第12.3.1节中强制绕流平板的例子的结果不同。可以看出,两个边界层相对厚度与普朗特数无关。

流函数的最大值反映了自然对流的泵效应,与 Pr 直接相关。Pr 的大值对应着修正流函数 f 的小值,修正流函数的最大值与总体积流量有关。欲解释给定流体的这种关系,必须对一维流函数进行计算:

$$\Psi(x,y)=4f(\eta)v\left(\frac{Gr_x}{4}\right)^{0\ 25}$$

通过计算可以看出,在相同的温差下,气体的体积流量要大于水的体积流量。

由于这种流动可以把能量带离物体表面,所以对于热流量的分析可以采用相同的方法,由下式确定:

$$q''_s=-\frac{k(T_s-T_\infty)}{x}\left(\frac{Gr_x}{4}\right)^{0\ 25}\frac{\mathrm{d}T^*}{\mathrm{d}\eta}\bigg|_{\eta=0}$$

计算出气体和水二者的热流量后,可以发现水的热流量比空气大100倍。这主要是由于热导、比热和密度共同决定了热流量的大小。在温度为300 K的常压下,水的导热系数大约是空气的30倍,比热为空气的4倍,密度为空气的1 000倍。虽然在空气中体积流量大,但是在水中热流量大。

12.3.3 印刷电路板的温度分布①

印刷电路板如图12.16所示,其上有4个平面固定的电子器件,发散出的能量大小在图中给出。电路板 y 轴垂直安装在支架上,在这种情况下传热系数 h 在 y 方向上是变化的。为了简化,假设在 x 方向上传热系数为一常数。为了近似描述 y 方向上传热系数的变化情况,在这里可以认为该板是由三块相邻的大小相等的板组成,三块板的传热系数各不相同,如图12.16所示。板和器件的热导系数 k 相同且与所在位置无关。为简单起见,假设4个器件具有相同的热导系数。

每块板和电子器件的特征方程为:

$$\nabla(k\nabla T)+q'''-H(T-T_\infty)=0$$

其中 q'''是每单位体积热源的功率(W/m^3),$H=2h/t$(W/m^{3}K),t 是板的厚度,T_∞是环境温度。由于板子的两面都有热对流发生,所以在 H 的定义式中有系数2。

采用PDE工具箱可以计算整块板的温度分布。首先,创建一个适当的绘图环境。进入Options菜单,设置Axis Limits为[0,17],两个轴的格间距为1,然后选择Snap。下一步,把Generic Scalar变为Heat Transfer。然后画出三个矩形和四个正方形,尺寸和位置如图12.16所示。结果生成设置表达式 $R1+R2+R3+SQ1+SQ2+SQ3+SQ4$。每个区域将由导热方程确定。

① 由 Yogendra Joshi, Department of Mechanical Engineering, University of Maryland, College Park, MD 提出。

然而，必须确定表 12.6 所示矩形区域的边界条件。因为每个 SQn 都全部存在于提前定义的区域(Rn)内，对于 PDE 工具而言并不产生新的边界，所以，设置表达式中的加号(+)并不具有创建边界的功能。注意 k, h, Q 的单位分别为 W/(cm K)、W/(cm^2 K)和 W/(cm^3)，与板子的尺寸单位一致。

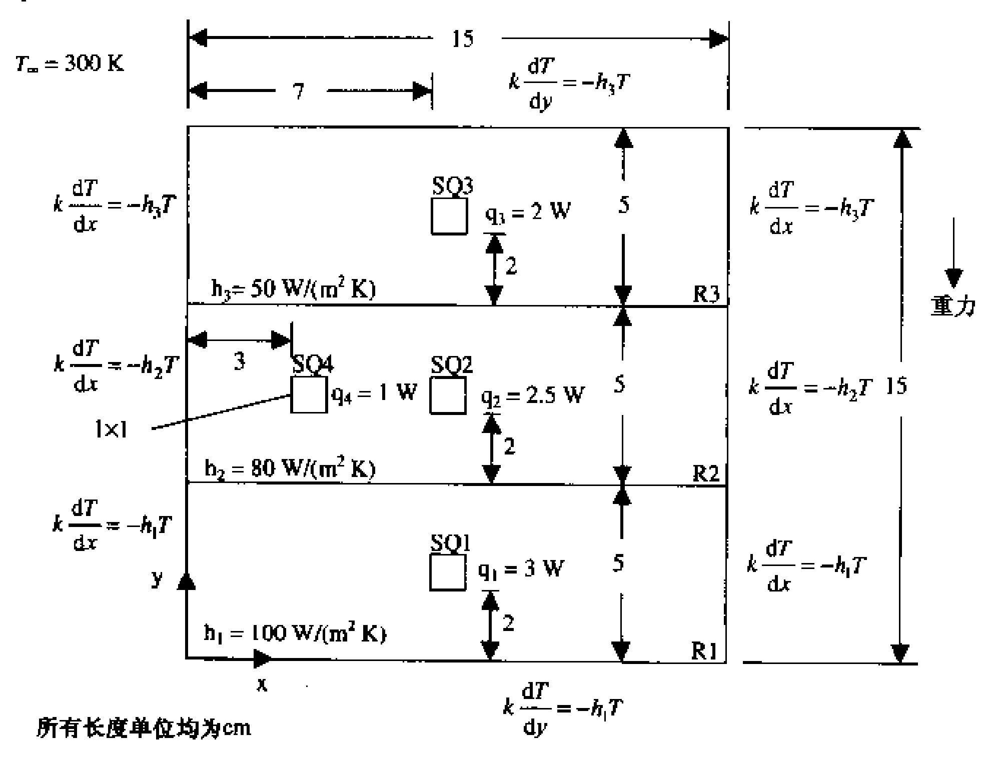


图 12.16　印刷电路板的几何尺寸和参数值

再下一步，进入 Boundary Mode，然后选择 Specify Boundary Conditions。表 12.6 给出的矩形边界条件是纽曼类型，这些边界都用带箭头的红线标出。双击每条线后，表 12.6 给出的值就会输入到相应的位置。然后选择 PDE Mode，把光标移到七个区域中的某一区域，并双击，显示 PDE Specification 窗口，表 12.6 给出的数值可通过该窗口输入。

表 12.6　印刷电路板的数据以及函数 pdetool 所使用的符号

	区域						
	R1	***R2***	***R3***	***SQ1***	***SQ2***	***SQ3***	***SQ4***
边界条件							
左							
g	0	0	0	—	—	—	—
$q \to h$(W/cm^2 K)	0.01	0.008	0.005	—	—	—	—
右							
g	0	0	0	—	—	—	—
$q \to h$(W/cm^2 K)	0.01	0.008	0.005	—	—	—	—
顶							
g	—	—	0	—	—	—	—
$q \to h$(W/cm^2 K)	—	—	0.005	—	—	—	—
底							
g	0	—	—	—	—	—	—
$q \to h$(W/cm^2 K)	0.01	—	—	—	—	—	—
PDE 说明							
$k \to k$(W/cm K)	0.003	0.003	0.003	0.003	0.003	0.003	0.003
$Q \to Q$(W/cm^3)	0	0	0	30	25	20	10
$h \to H$(W/cm^3 K)	0.1	0.08	0.05	0.1	0.08	0.05	0.08
$Text \to T_\infty$(K)	300	300	300	300	300	300	300

对网格初始化并优化，然后得出结果。绘制出温度分布图和热流量图，如图 12.17 和图 12.18 所示。

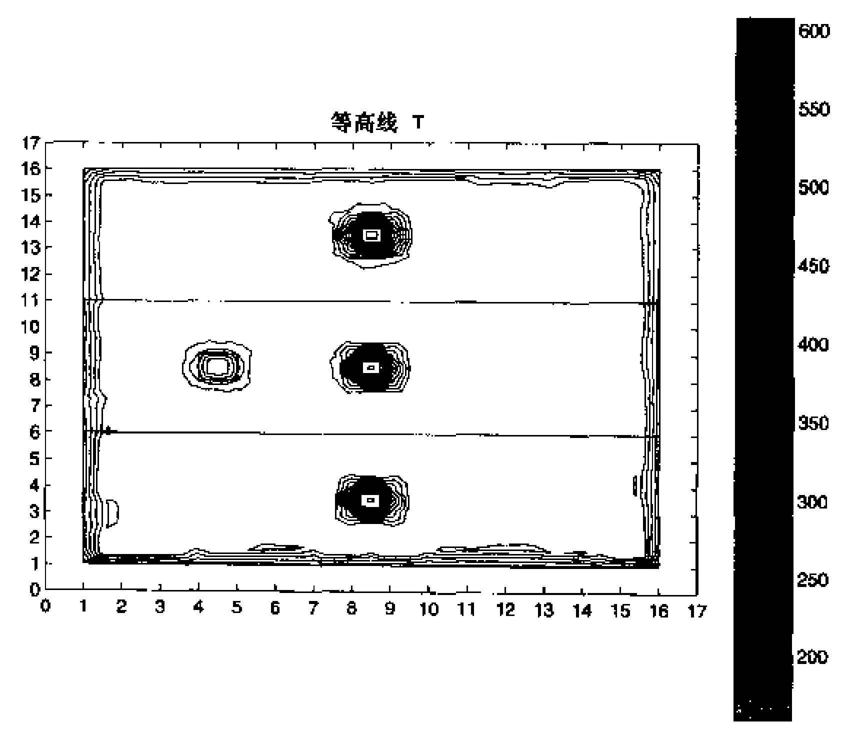


图 12.17 图 12.16 所示印刷电路板的温度分布

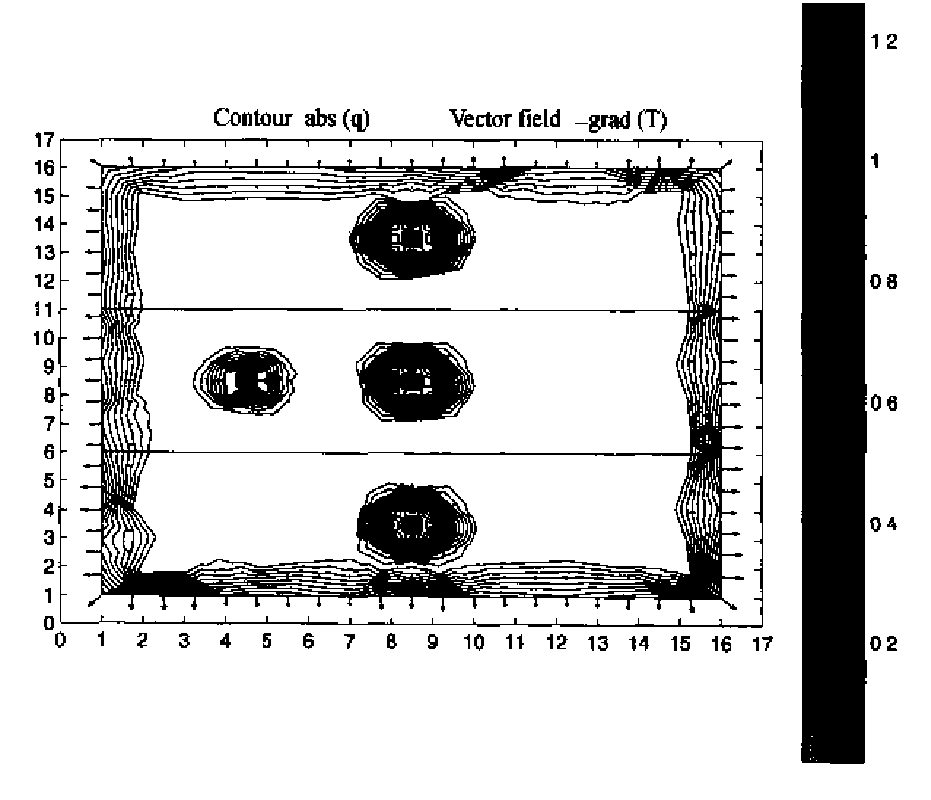


图 12.18 图 12.16 所示印刷电路板的热流量

12.4 辐射传热

12.4.1 辐射角系数——平行平面上的面积微元到任意矩形

当分析扩散表面包壳的热辐射时需要计算辐射角系数。在许多计算角系数的方法中较常用的是根据角系数的定义式对其进行求解。首先考虑面积微元 dA_1 和一个有限面积 A_2 间角系数的一般表达式：[①]

$$\mathrm{d}F_{2-d_1} = \frac{\mathrm{d}A_1}{A_2}\int_{A_2}\frac{\cos\theta_1\cos\theta_2}{\pi S^2}\mathrm{d}A_2 \tag{12.2}$$

式中 S 为 $\mathrm{d}A_1$ 到 A_2 上某一点的距离，如图 12.19 所示。角 $\theta_j, j=1,2$ 是 S 与表面法线的夹角。角系数互换性为：

$$A_2\mathrm{d}F_{2-d_1} = \mathrm{d}A_1 F_{d_1-2} \tag{12.3}$$

因此，F_{d_1-2}可以表示为：

$$F_{d_1-2} = \int_{A_2}\frac{\cos\theta_1\cos\theta_2}{\pi S^2}\mathrm{d}A_2 \tag{12.4}$$

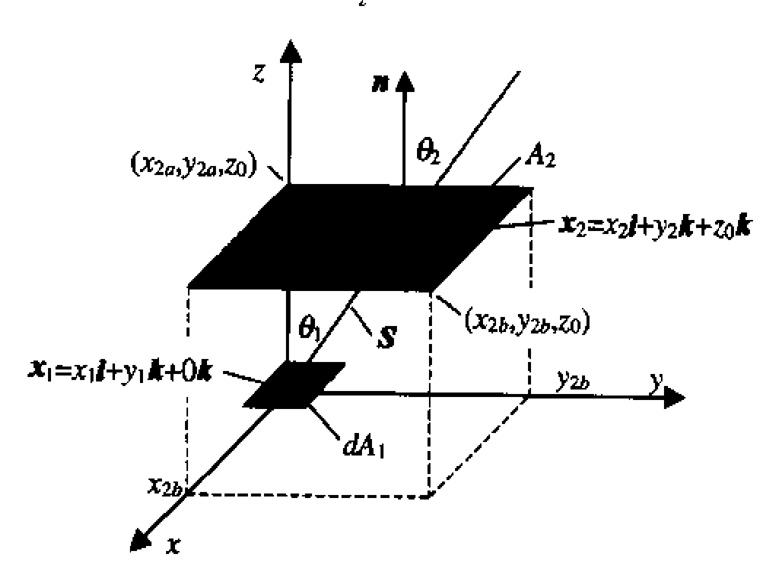


图 12.19 面积微元和有限矩形平行分布的几何图形

然后进行数值积分。假设 $\mathrm{d}A_1$ 和 A_2 相互平行且 A_2 是一个矩形，通过适当的程序设计可以把这些限制条件去掉。在这个例子中，方程(12.4)可以改写为：

$$F_{d_1-2} = \frac{1}{\pi}\int_{y_{2a}}^{y_{2b}}\int_{x_{2a}}^{x_{2b}}\frac{\cos\theta_1\cos\theta_2}{S^2}\mathrm{d}x_2\mathrm{d}y_2$$

式中向量 S 为：

① 参见 R Siegel 和 J R Howell, *Thermal Radiation Heat Transfer*, 第三版, Hemisphere Pub, Washington, 1992 年, 第 189－252 页。

$$\boldsymbol{S} = \boldsymbol{x}_2 - \boldsymbol{x}_1 = (x_2 - x_1)\boldsymbol{i} + (y_2 - y_1)\boldsymbol{j} + (z_2 - z_1)\boldsymbol{k}$$

和

$$S = |\boldsymbol{S}|$$

角度可以由向量 S 和矩形上的法线向量 $\boldsymbol{n} = \boldsymbol{k}$ 表示：

$$\cos\theta_1 = \cos\theta_2 = \frac{\boldsymbol{n}\cdot\boldsymbol{S}}{|\boldsymbol{S}|}$$

由于两个平面相互平行，式中 $\theta_1 = \theta_2 = \theta$。所以：

$$F_{d_1-2} = \frac{1}{\pi}\int_{y_{2a}}^{y_{2b}} I_{x2}(y_2)\mathrm{d}y_2$$

式中，

$$I_{x2}(y_2) = \int_{x_{2a}}^{x_{2b}} f(x_2, y_2)\mathrm{d}x_2$$

且

$$f(x_2, y_2) = \frac{\cos^2\theta}{|\boldsymbol{S}|^2} = \frac{(\boldsymbol{n}\cdot\boldsymbol{S})^2}{|\boldsymbol{S}|^4}$$

首先创建函数 kernel 来计算 A_2 上任意一点的 $f(x_2, y_2)$。该函数的输入值为 A_2 上一点的坐标 x_2、y_2 和 $z_2 = z_0$。该函数用向量形式确定长度 S 和 $\cos\theta$。注意为了与函数 quad8 保持一致，kernel 函数必须返回一个向量，该向量的维数必须与输入向量 $\boldsymbol{x}$ 的维数相等。这样函数 quad8 既能减少被积函数的调用次数，同时又能提供所需的数据。kernel 函数通过用 length 函数确定 x 的元素个数来实现此项功能。

```
function f = kernel(x,y,z,xpl)
L = length(x);
S = [x - xpl(1);ones(1,L) * y - xpl(2);ones(1,L) * z - xpl(3)];
f = dot(repmat([0 0 1]',1,L),S).^2./dot(S,S).^2;
```

现在创建一个函数 Intx2，它能在给定 y 值时，由 quad8 来对 x 进行积分。当公式中有二重积分时，quad8 调用 Intx2 函数，因此，在 Intx2 中，变量 y 必须为一任意维的向量。和函数 kernel 一样，Intx2 计算的向量维数必须与输入 y 的维数相等。

```
function Int = Intx2(y,xp1,x2a,x2b,z)
Int = zeros(1,length(y));
for i = 1:length(y)
  Int(i) = quad8('kernel',x2a,x2b,[],[],y(i),z,xp1);
end
```

在下面的两个例子中将用到这些函数。表 12.7 给出了这两个例子中的几何尺寸和计算出的角系数。利用下面给出的程序可以获得表中两个角系数：

```
x21 = -1;y21 = -1;
x22 = 0;y22 = 0;
x1 = [0 0 0];
Fd12Set1 = quad8('Intx2',y21,y22,[],[],x1,x21,x22,5)/pi
x22 = 1;y22 = 1;
Fd12Set2 = quad8('Intx2',y21,y22,[],[],x1,x21,x22,1)/pi
```

此外，可以给出表 12.7 中组 #1 的角系数随距离变化的曲线，程序为：

```
x21 = -1;y21 = -1;
x22 = 0;y22 = 0;
x1 = [0 0 0];nz = 20;
z = linspace(0.1,5,nz);
for i = 1:nz
 Fd12(i) = quad8('Intx2',y21,y22,[],[],x1,x21,x22,z(i))/pi;
end
plot(z,Fd12)
xlabel('Separation of surfaces')
ylabel('View factor')
text(1.8,.2,'Areas are in parallel planes')
text(1.8,.19,'Finite area is 1 x 1(arbitrary length units)')
text(1.8,.18,'Differential area is aligned with a corner')
title('Radiation view factor - differential to finite area')
```

执行结果如图 12.20 所示。当两个平行平面距离趋于 0 时，F_{d_1-2}趋于极限值 0.25。这是因为点 dA_1 与 A_2 的一个顶角在同一条直线上，所以当这两个平行平面相互靠近时，A_2 切掉了 dA_1 的 1/4。在组 #2 的几何尺寸中，dA_1 和 A_2 的中心点在同一直线上，当两个平行平面相互靠近时，F_{d_1-2}的极限值为 1.0。

表 12.7 实例的几何尺寸与角系数值

参数	组 #1	组 #2
A_2 的几何尺寸		
原点的 X 坐标 x_{2a}	-1	-1
原点的 Y 坐标 y_{2a}	-1	-1
顶点的 X 坐标 x_{2b}	0	1
顶点的 Y 坐标 y_{2b}	0	1
两平面间的距离 z_0	5	1
角系数的计算值 F_{d_1-2}	0 012 1	0.554 1

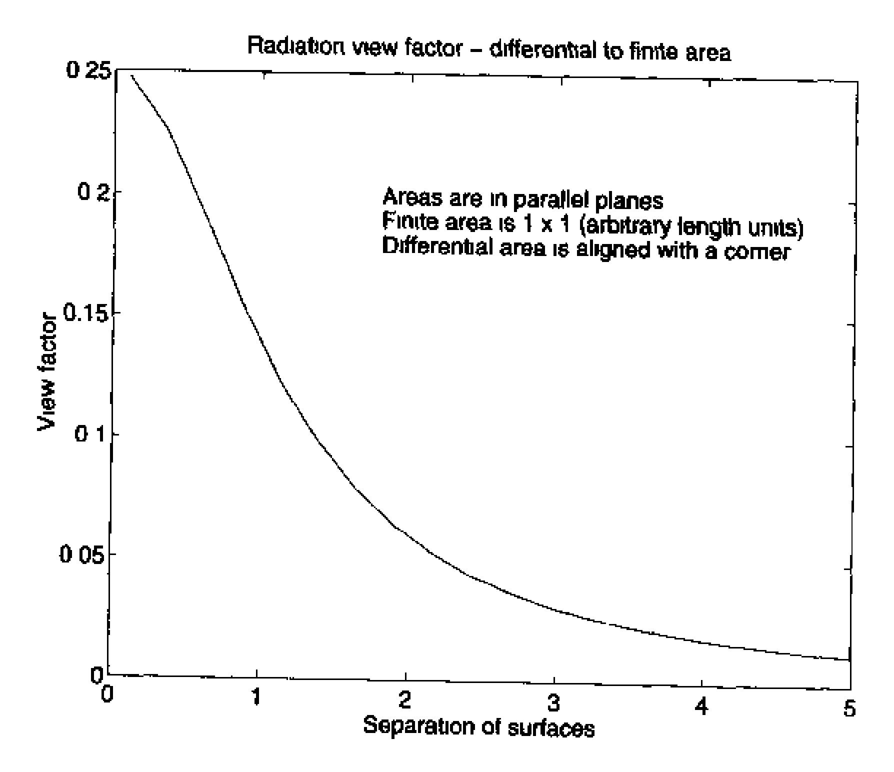


图 12.20 图 12.19 所示两个平行平面间距离与结构参数的对应关系

值得注意的是函数 dblquad 可以计算二重积分，但它没有输入参数，而这些参数又是公式需要的。

12.4.2 平行平面上两个矩形间的角系数

先前计算的是从面积微元到有限面积之间的角系数。通过对面积微元积分，可以得到两个面积为有限的平面间的角系数。角系数由下式定义：

$$F_{2-1} = \frac{1}{\pi A_2}\int_{A_1}\int_{A_2} \frac{\cos\theta_1\cos\theta_2}{S^2}\mathrm{d}A_2\mathrm{d}A_1 = \frac{1}{\pi A_2}\int_{y_{1a}}^{y_{1b}}\int_{x_{1a}}^{x_{1b}}\int_{y_{2a}}^{y_{2b}}\int_{x_{2a}}^{x_{2b}} \frac{\cos\theta_1\cos\theta_2}{S^2}\mathrm{d}x_2\mathrm{d}y_2\mathrm{d}x_1\mathrm{d}y_1 \quad (12.5)$$

式中的变量定义与前面一节所讨论的变量定义相似。图 12.21 定义了式中四重积分的积分限。有许多方法可以计算四重积分，在此使用直接积分法。两个平面平行，其边缘也平行，式(12.5)可以写为：

$$F_{2-1} = \frac{1}{\pi A_2}\int_{y_{1a}}^{y_{1b}} I_{x1}(y_1)\mathrm{d}y_1$$

其中，

$$I_{x1}(y_1) = \int_{x_{1a}}^{x_{1b}} I_{y_2}(x_1, y_1)\mathrm{d}x_1$$

$$I_{y_2}(x_1, y_1) = \int_{y_{2a}}^{y_{2b}} I_{x2}(x_1, y_1, y_2)\mathrm{d}y_2$$

$$I_{x2}(x_1, y_1, y_2) = \int_{x_{2a}}^{x_{2b}} f(x_1, y_1, x_2, y_2)\mathrm{d}x_2$$

$$f(x_1, y_1, x_2, y_2) = \frac{\cos^2\theta}{|\boldsymbol{S}|^2} = \frac{(\boldsymbol{n}\cdot\boldsymbol{S})^2}{|\boldsymbol{S}|^4}$$

$$\boldsymbol{S} = (x_1 - x_2)\boldsymbol{i} + (y_1 - y_2)\boldsymbol{j} + (z_1 - z_2)\boldsymbol{k}$$

并且，$\boldsymbol{n} = \boldsymbol{k}$，$z_1 = 0$，$z_2 = z_0$。

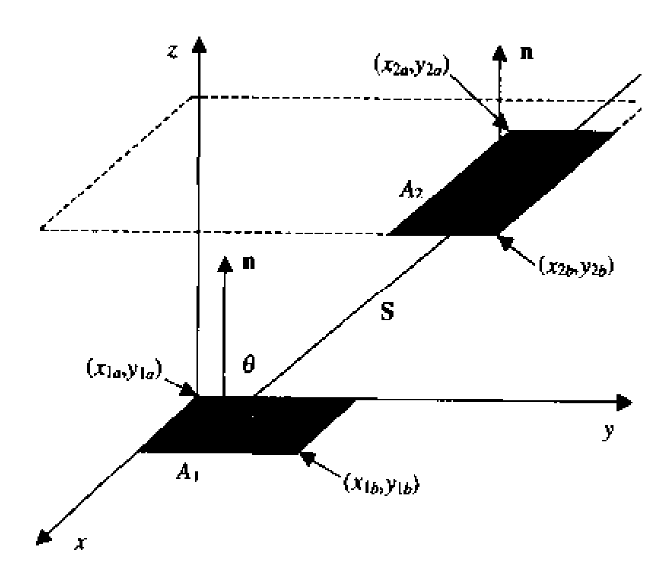


图 12.21 确定两个平行矩形之间角系数的几何图形

虽然可以采用直接积分的方法，但由于 MATLAB 在数值积分算法中采用了自适应的方式，所以直接积分计算时间太长。

计算四重积分需要用四个函数。其中两个函数已经在第 12.4.1 节创建，分别是 kernel 和 Intx2，其中 Intx2 调用函数 kernel。现在再来创建另外两个函数，首先创建函数 Inty2 用于计算 I_{y2}且可以调用 Intx2。

```
function Int = Inty2(x1,y1,XY2,z)
Int = zeros(1,length(x1));
for i = 1:length(x1);
  xp1 = [x1(i)y1 0];
  Int(i) = quad8('Intx2',XY2(1,2),XY2(2,2),[],[],xp1,XY2(1,1),XY2(2,1),z);
end
```

第二个要创建的函数为 Intx1，用来计算 I_{x1}，且可以调用 Intx2。

```
function Int = Intx1(y1,x1a,x1b,XY2,z)
Int = zeros(1,length(y1));
for i = 1:length(y1);
 Int(i) = quad8('Inty2',x1a,x1b,[],[],y1(i),XY2,z);
end
```

然后就可以计算平行平面上任意矩形的角系数，计算角系数所需的数据由表 12.8 组 #1 给出。计算组 #2 的角系数的方法同上，所不同的是数据变了。程序为：

```
x1a = -1;y1a = -1;x2a = -1;y2a = -1;
x1b = 1;y1b = 1;x2b = 1;y2b = 1;
z = 2;
A2 = abs(x2a - x2b) * abs(y2a - y2b);
XY2 = [x2a,y2a;x2b,y2b];
Ff = quad8('Intx1',y1a,y1b,[],[],x1a,x1b,XY2,z)/A2/pi
```

表 12.8 计算角系数所需数据

参数	组 #1	组 #2
A_1 的几何尺寸		
原点的 X 坐标，面积 1，x_{1a}	-1	-2
原点的 Y 坐标，面积 1，y_{1a}	-1	-2
顶点的 X 坐标，面积 1，x_{1b}	1	0
顶点的 Y 坐标，面积 1，y_{1b}	1	0
A_2 的几何尺寸		
原点的 X 坐标，面积 2，x_{2a}	-1	2
原点的 Y 坐标，面积 2，y_{2a}	-1	2
顶点的 X 坐标，面积 2，x_{2b}	1	0
顶点的 Y 坐标，面积 2，y_{2b}	1	0
平面间的距离	2	2
角系数计算值(F_{2-1})	0.199 8	0.043 3

12.4.3 具有漫射、灰体壁面的包壳辐射

考虑热辐射时，经常遇到的问题是要根据包壳的辐射来确定温度和传热率，该包壳具有漫

射和灰体表面而且包含一种不参加辐射的媒介物。这种情况发生在炉子、房间和其他被包围起来的空间里。采用漫射灰体表面的假设,使其和一般辐射模式相比具有较小的复杂性。漫射说明所有表面的辐射和吸收强度与方向无关,灰度说明发射率和吸收率与波长无关。然而,即使这样简化,包壳的问题仍需要用相当大的精力去解决。这些问题可以很自然地在矩阵中表现出来,因此,MATLAB 提供了一个理想的环境对其进行系统阐述并予以解决。由此分析得出下列公式:[①]

$$\frac{Q_k}{A_k} = q_k = \frac{\varepsilon_k}{1-\varepsilon_k}(\sigma T_k^4 - q_{0,k}) \tag{12.6}$$

$$\frac{Q_k}{A_k} = q_k = q_{0,k} - \sum_{j=1}^{N} F_{k-j} q_{0,j} = \sum_{j=1}^{N} F_{k-j}(q_{0,k} - q_{0,j}) \tag{12.7}$$

其中 Q_k 是表面 k 的传热率。A_k 为面积,q_k 是热流量,$q_{o,k}$是辐射系数,F_{k-j}为从 k 表面辐射出去而被 j 表面接收的能量的角系数,N 为包壳的表面个数,$\delta = 5.67 \times 10^{-8}$ W/(m^2 K^4)是斯蒂芬-玻耳兹曼常数。该公式假设每个表面接收和发射的辐射作为一整体,其强度与方向无关。对于一个给定的问题,可以通过对包壳表面进行细分直到获得的结果与面积分割无关,从而对该假设进行计算。

对于一般的包壳问题,必须指定每个表面的传热率或温度。一旦指明后,式(12.6)和式(12.7)就确定了每个表面惟一独立的关系。如果指定了温度,则式(12.6)和式(12.7)相等,表示为:

$$q_{0,k} - \sum_{j=1}^{N} F_{k-j} q_{0,j} = \frac{\varepsilon_k}{1-\varepsilon_k}(\sigma T_k^4 - q_{0,k}) \tag{12.8}$$

如果指定了传热率,则式(12.8)可以改为:

$$\frac{Q_k}{A_k} = q_{0,k} - \sum_{j=1}^{N} F_{k-j} q_{0,j} \tag{12.9}$$

写成矩阵形式:

$$\begin{bmatrix} d_1 - F_{1-1} & -F_{1-2} & \cdots & -F_{1-N} \\ -F_{2-1} & d_2 - F_{2-2} & & -F_{1-2} \\ \vdots & & & \vdots \\ -F_{N-1} & -F_{N-2} & & d_N - F_{N-N} \end{bmatrix} \begin{bmatrix} q_{0,1} \\ q_{0,2} \\ \vdots \\ q_{0,N} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix}$$

当指定了温度时,式中,

$$d_k = 1/(1-\varepsilon_k)$$

$$b_k = \frac{\varepsilon_k}{1-\varepsilon_k}\sigma T_k^4$$

当指定了传热率时,式中,

$$d_k = 1$$

$$b_k = \frac{Q_k}{A_k}$$

矩阵中每一行代表一个表面,其形式取决于该表面是指定温度还是指定传热率。对以上

① R.Siegel 和 J.R.Howell,出处同上,第 189～252 页。

任何一种情况，表面的辐射系数都是未知的。因此，方程组中包含 N 个方程和 N 个未知辐射系数。一旦辐射系数被确定，对于给定表面未知的温度和传热率就可由式(12.6)和式(12.7)确定。除特殊情况外，用式(12.6)计算很简单，当发射率为1时，则应使用式(12.7)。

利用下例说明计算结果。对于一个横截面为矩形的炉子，假设可无限延伸到纸面内部。图12.22定义了它的几何尺寸。表12.9定义了其角系数，这些系数可以通过霍特尔(Hottel)的 crossed - string 方法计算[①]。表中粗体印刷的值是待选值，在此基础上，对角系数进行代数运算求出其他值。

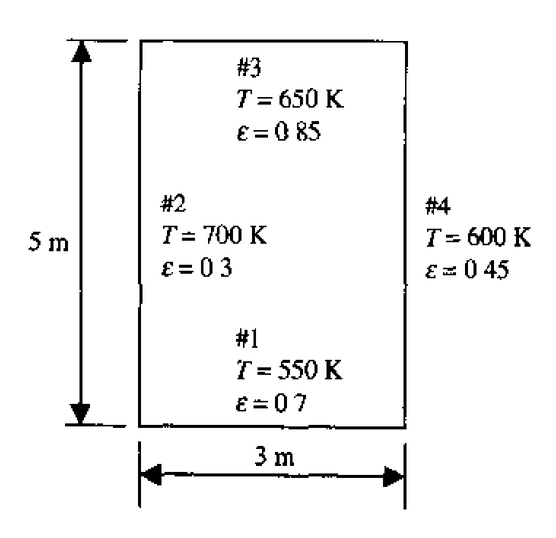


图12.22 具有灰度侧壁的包壳辐射时几何尺寸和表面特性

表 12.9 图 12.22 所示包壳的角系数 F_{i-j}

i/j	1	2	3	4
1	**0**	**0.361 5**	**0.277 0**	0 361 5
2	0 216 9	**0**	**0.216 9**	0 566 2
3	0 277 0	0 361 5	**0**	0 361 5
4	0 216 9	0 566 2	0 216 9	0

下面的程序利用了标识符 $c_k, k = 1,2,\cdots N$，当给定温度时，$c_k = 0$；当给定传热率时，$c_k = 1$。然后根据 c_k 选择 d_k 和 b_k 的值。此外所有的向量维数必须为 N 且矩阵 $\boldsymbol{F}$ 为 $(N \times N)$ 阶。程序为：

```
sigma = 5.6693E - 8;N = 4;
A = [3 5 3 5];epsilon = [0 7 0 3 0.85 0.45];
T = [550 700 650 600];
F = - [0 0.3615 0.277 0.3615;. . .
       0.2169 0 0.2169 0.5662;. . .
       0.277 0.3615 0 0.3615;. . .
       0.2169 0.5662 0.2169 0];
Q = [0 0 0 0];
```

① 参见 R Siegel 和 J R Howell，出处同上。

```
c = [0 0 0 0];
b = sigma * epsilon ./(1 - epsilon) .* (1 - c) .* T.^4 + c .* Q ./A;
d = (1 - c) .* 1 ./(1 - epsilon) + c;
for k = 1:N
  F(k,k) = d(k) + F(k,k);
end
q0 = F \ b';
Q = A .* epsilon./(1 - epsilon) .* (1 - c) .* (sigma * T.^4 - q0')
T = c .* ((Q./A .* (1 - epsilon)./epsilon + q0')/sigma).^(1/4)
q = Q./A
```

执行程序后得到:

$Q = [-8\,627.9, 8\,061.1, 4\,525.9, -3\,959.9]$W, $q = [-2\,876, 1\,612.2, 1\,508.6, -791.8]$W/m^2。

可以看出,传热率 Q 之和接近于零。

12.4.4 炉内热平板的瞬态热辐射[①]

假设有一垂直悬挂在炉内的平板。炉子的一内壁与含有热元件的平板表面平行,炉子和平板的温度最初都为室温。为了使平板的温度在 t_h 时间内上升为 T_e,加热元件所需的燃烧热能 Q 可由平板和炉壁内能量平衡关系确定,从而得出下式:

$$\frac{dT_w}{dt} = P_1 Q - P_2(T_w^4 - T_p^4)$$

$$\frac{dT_p}{dt} = -P_3(T_p^4 - T_w^4)$$

其中平板和炉内壁作为一个整体考虑,T_w 是炉内壁的温度,T_p 是平板的温度。如果把这种结构假设为具有两个漫射灰度表面的包壳,则[②]:

$$P_1 = \frac{1}{m_w c_w}$$

$$P_2 = \frac{\sigma}{m_w c_w}\left[\frac{1-\varepsilon_p}{\varepsilon_p A_p} + \frac{1}{A_w F_{wp}} + \frac{1-\varepsilon_w}{\varepsilon_w A_w}\right]^{-1}$$

$$P_3 = \frac{\sigma}{m_p c_p}\left[\frac{1-\varepsilon_p}{\varepsilon_p A_p} + \frac{1}{A_w + F_{pw}} + \frac{1-\varepsilon_w}{\varepsilon_w A_w}\right]^{-1}$$

其中,m_p 和 m_w 分别为平板和炉内壁的质量;c_p 和 c_w 分别为平板和炉内壁的比热;ε_w 和 ε_p 分别为平板和炉内壁的辐射率,A_p 和 A_w 分别为平板和炉内壁的面积;F_{pw} 和 F_{wp} 分别为平板和炉内壁的角系数;$\sigma = 5.67\times10^{-8}$ W/(m^2 K^4)是斯蒂芬-玻耳兹曼常数。

对于某结构,假设 $P_1 = 1.67\times10^{-5}$ K/J、$P_2 = 8.8\times10^{-14}$ s^{-1}K^{-3} 和 $P_3 = 6.3\times10^{-13}$ s^{-1}K^{-3}。然后确定使平板温度在时间 $t_h = 10$ 分钟(600 s)内上升到 $T_e = 1\,100$ K 所需的 Q 值。假设平板和炉子的温度初始值为 300 K。首先创建两个函数,一个被函数 ode45 用来求解两个一阶微分方程,称为 RadTemp 函数;另一个由函数 fzero 用来确定 Q 值,称为 *QGen* 函数。

---

① Yogendra Joshi, Department of Mechanical Engineering, University of Maryland, College Park, MD 提出。

② 参见 F P. Incropera 和 D P. DeWitt,出处同上,第 13 章。

对于函数 RadTemp，有：

```
function dTdt = RadTemp(t,T,flag,P1,P2,P3,Q)
dTdt = [P1 * Q - P2 * (T(1)^4 - T(2)^4); - P3 * (T(2)^4 - T(1)^4)];
```

其中：$T(1) = T_w(t)$，$T(2) = T_p(t)$。对于函数 QGen，有：

```
function PlateTempDev = QGen(Q,Te,th,P1,P2,P3,T1o,T2o,tend)
[t,T] = ode45('RadTemp',[0,tend],[T1o;T2o],[],P1,P2,P3,Q);
PlateTempDev = Te - interp1(t,T(:,2),th,'spline');
```

其中：$T1o = T_w(0)$，$T2o = T_p(0)$，用函数 interp1 来确定温度 $T_e = Te = 1\ 100$ K 时的时间。下面的程序确定 Q 值并画出炉内壁和平板温度随时间变化的曲线：

```
P1 = 1.67e-5; P2 = 8.8e-14; P3 = 6.3e-13; Qguess = 100000;
Te = 1100; th = 600; tend = 660; T1o = 300; T2o = 300;
options = optimset('display','off');
Q = fzero('Qgen',Qguess,options,Te,th,P1,P2,P3,T1o,T2o,tend);
[t,T] = ode45('RadTemp',[0,tend],[T1o;T2o],[],P1,P2,P3,Q);
plot(t,T(:,1),'k-',t,T(:,2),'k--')
z = axis;
hold on
plot([0,z(2)],[Te,Te],'k',[th,th],[z(3),z(4)],'k')
xlabel('Time(s)')
text(0.05 * z(2),0.85 * z(4),['Q = 'num2str(Q,6)'W'])
ylabel('Temperature(K)')
legend('Wall temperature','Plate temperature',2)
```

程序执行后产生图 12.23。

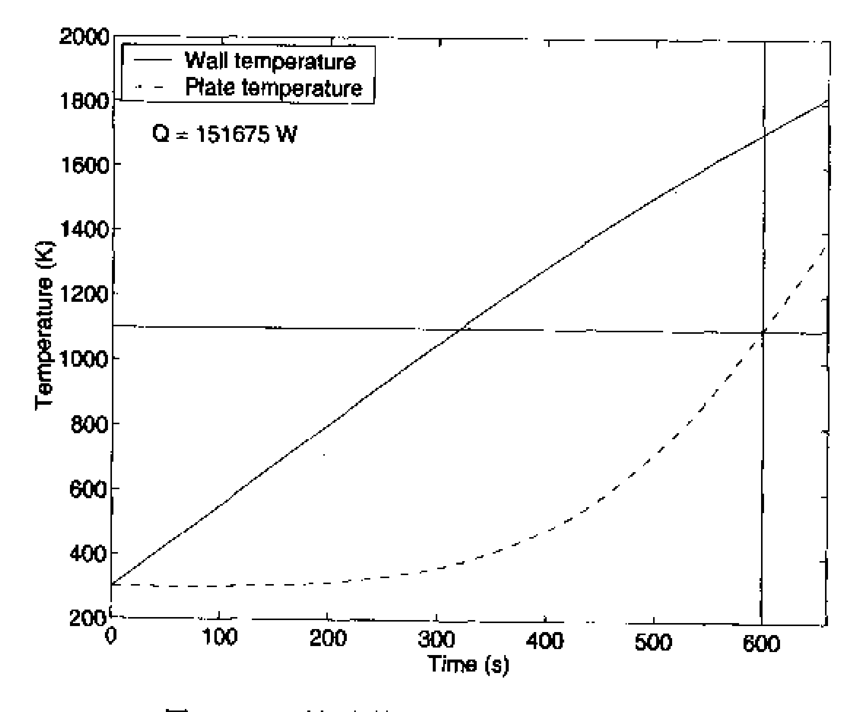


图 12.23　平板和炉壁温度随时间变化的曲线图

练习

12.1 一维大平壁导热可以表示为：

$$k(T)\ \frac{\mathrm{d}T}{\mathrm{d}x}=q$$

其中 q 为热流量，$k(T)$是依赖于温度的导热系数，T 是温度，x 是空间坐标。设侧壁由绝缘矿渣绒构成，导热系数根据下式变化：

$$k(T)=-A+BT \qquad 240\ \mathrm{K}<T<365\ \mathrm{K}$$

其中 $A=0.408$，$B=0.00032$，k 的单位为 W/m·K，T 的单位为 K。当热流量 $q=12.5\ \mathrm{W/m^2}$，侧壁厚 0.1 m，并且 $x=0.1$ m 处侧壁的表面温度为 300 K 时，确定 $x=0.1$ m 的温度。计算结果与下式做比较：

$$A(T-T_1)+\frac{B}{2}(T^2-T_1^2)=qx$$

其中 $T(0)=T_1$。根据导热系数的平均值计算热流量，结果与真实值非常接近。

12.2 为了估计加热(冷却)牛奶的时间，一个标准的塑料牛奶瓶可以看做一个集总热容。这种条件下的特征方程可以表示为：

$$\frac{\mathrm{d}T}{\mathrm{d}t}=\frac{Q}{mc_v}$$

式中，$Q=hA(T_{amb}-T)$是周围环境向牛奶瓶进行的热传递，m 是瓶子的质量，c_V 是瓶子的比热。对于一个简单的辐射模型为：

$$Q=A\sigma\varepsilon(T^4-T_{amb}^4)$$

式中 $\sigma=5.667\times10^{-8}\ \mathrm{W/m^2\cdot K^4}$，是斯蒂芬-玻耳兹曼常数，$\varepsilon$ 是发射率，A 是表面积，T 是瓶子的温度，T_{amb}是环境温度。

确定该系统在具有辐射和无辐射条件下的时间常数。时间常数 τ 是指瓶子和周围环境的温差从初始值下降到 63.2% 所用的时间，由下式表示：

$$\frac{T_{amb}-T(\tau)}{T_{amb}-T(0)}=0.368$$

在没有辐射的情况下，设瓶子表面的自然对流传热系数 $h=2\ \mathrm{W/K\cdot m^2}$，$T_{amb}=30$ ℃，$T(0)=5$ ℃。在有辐射的情况下，设 $\varepsilon=0.5$，$m=3.5$ kg，$c_v=4.2$ J/g·K，$A=0.3\ \mathrm{m^2}$，$T(0)=5$ ℃，$T_{amb}=30$ ℃。

[答案：在无辐射的情况下：$\tau=6.8$ hr$(=mc_v/(hA))$；在有辐射的情况下：$\tau=2.78$ hr，是初始温度的函数。]

12.3 绝热管子的绝热层外表面通过对流损失的热量由下式决定：

$$q=\frac{2\pi L(T_i-T_\infty)}{\frac{1}{k}\ln(r_o/r_i)+\frac{1}{r_oh}}$$

式中 L 是管子的长度，r_o 是绝热层的外径，r_i 是绝热层的内径，k 是导热系数，h 是传热系数。当 r_o 取值较小时，附加的绝热层可以提高传热率。为了描述这种效果

把 q 作为 r_o 的函数绘制出曲线，其中 r_o 的间隔为 $r_o = k/h$。设 $h = 5\ \text{W/m}^2\text{K}, k = 0.1\ \text{W/m·K}, r_i = 0.01\ \text{m}, L = 1\ \text{m}$。一般认为增加绝热层会降低传热率，但这个例子是一个例外。

12.4 温度传感器可以用于测量流动液体的温度。该传感器装在一个小尺寸的圆柱探头上，而该探头沿管壁垂直于流体流向伸入流体中。可以把探头视为肋片，其上的温度分布为：

$$\frac{T(x) - T_\infty}{T_b - T_\infty} = \frac{\cosh m(L - x)}{\cosh mL}$$

其中，

$$m^2 = \frac{hP}{kA_c}$$

T_b 是侧壁的温度，T_∞ 是流体的温度，$P = \pi d$ 是直径为 d 的探头周长，$A_c = \pi d^2/4$ 是横截面面积，k 是热导，h 是传热系数，L 是探头长度。

由于沿着探头有对流发生所以探测的温度误差为：

$$e = T(L) - T_\infty = \frac{T_b - T_\infty}{\cosh ml}$$

绘出误差与探头长度的关系曲线，其中 $0.005 \leqslant L \leqslant 0.1\ \text{m}$，$k$ 取 $20 \leqslant k \leqslant 400\ \text{W/m·K}$ 中的几个值。

设流体的温度为 100 ℃，侧壁的温度为 80 ℃，流体和探头的传热系数为 25 $\text{W/m}^2\text{·K}$。

12.5 由于存在非线性自然热辐射，所以在辐射路径上隔热板会降低传热效率，因此在需要绝热的情况下，采用隔热板是很重要的。隔热板的作用可以理解为两个无限大平板之间的热传递，这两个平板温度分别为 T_1 和 T_2 且两板间为真空。当表面像黑体一样辐射时，在没有隔热板的情况下传热率 q 为：

$$q = \sigma(T_1^4 - T_2^4)$$

当有一个隔热板时，传热率由下式决定：

$$q = \sigma(T_1{}^4 - T_m{}^4)$$
$$q = \sigma(T_m^4 - T_2{}^4)$$

其中 T_m 是隔热板的温度，当有两个隔热板时，传热率 q 为：

$$q = \sigma(T_1{}^4 - T_{m1}{}^4)$$
$$q = \sigma(T_{m1}{}^4 - T_{m2}{}^4)$$
$$q = \sigma(T_{m2}{}^4 - T_2{}^4)$$

其中 $\sigma = 5.667 \times 10^{-8}\ \text{W/m}^2\text{·K}^4$ 是斯蒂芬-玻耳兹曼常数。T_{m1}和 T_{m2}分别为隔热板 1 和隔热板 2 的温度。如果这两个平板的温度分别为 100 ℃和 20 ℃，则分别确定在没有隔热板、有一个隔热板和有二个隔热板情况下的传热率。

[答案：没有隔热板时：$q = 680\ \text{W/m}^2$；有一个隔热板时：$q = 340.1\ \text{W/m}^2$，$T_m = 340.15\ \text{K}$；有两个隔热板时：$q = 226.7\ \text{W/m}^2$，$T_{m1} = 352.2\ \text{K}$，$T_{m2} = 326.7\ \text{K}$。]

12.6 普朗克分布表示在某一温度下黑体辐射的单色辐射率，由下式决定：

$$E_{\lambda,b}(\lambda,T)=\frac{C_1}{\lambda^5[\exp(C_2/\lambda T)-1]}$$

式中 λ 是波长，T 是温度，$E_{\lambda,b}$ 是单色辐射率，$C_1=3.742\times10^8$ W.μm^4/m^2，$C_2=1.439\times10^4$ μm·K。在辐射计算中，通常需要对该函数在一定波长范围内进行积分。当所有波长进行积分时，就得到下式：

$$\int_0^\infty E_{\lambda,b}\mathrm{d}\lambda=\sigma T^4$$

式中，$\sigma=5.667\times10^{-8}$ W/m^2·K^4 是斯蒂芬-玻耳兹曼常数。在 $T=300$ K、400 K 和 500 K 时，分别对上式进行数值积分并将结果与精确值比较。注意：数值积分中给出上下限有相当大的困难。对于下限可以用一个非常小的非零数来近似，例如取 0.5 μm。对于上限，其取值主要依赖于 T 值，例如可以取 150 μm。

12.7 用第 12.3.1 节介绍的方法来确定当 $Pr=0.07$、0.7 和 7.0 时温度边界层和速度边界层的厚度。在第 12.3.1 节中，边界层温度分布 δ_T 被定义为 η 的值，$y_4=T^*(\eta=\delta_T)=0.99$；边界层速度分布 δ_u 被定义为 η 的值，$y_2=u(\eta=\delta_u)=0.99$，对于每个普朗特数，用 $Pr^{1/3}$ 的关系比较 δ_u/δ_T 与预测值的差异。

[答案见表 12.10。]

表 12.10　练习 12.7 的答案

Pr	δ_u	δ_T	δ_L/δ_T	$Pr^{1/3}$
0 07	5 052 6	15 026 1	0.336 3	0.412 2
0.7	4 968 4	5 724 3	0.867 9	0 887 9
7 0	4 917 7	2 504 6	1 963 5	1 912 8

12.8 空气以 $u_\infty=1$ m/s 的速度流过平板，$Pr=0.7$，$v_{\text{ms}}=1.5\times10^{-5}$ m/s^2。用第 12.3.1 节给出的公式计算 $T(x,y)$，区间为 $0<x<x_{\text{cru}}$ 和 $0<y<10x_{\text{cru}}/(\text{Re}_{\text{cru}})^{1/2}$，其中，

$$\text{Re}_{\text{cru}}=\frac{u_\infty x_{\text{cru}}}{v_{\text{ius}}}=5\times10^{-5}$$

绘出 $T(x,y)$ 在这一区域上的等高线图。在第 12.3.2 节中 $T(x,y)=y_4$，且：

$$\eta=y\sqrt{u_\infty/v_{\text{ius}}x}$$

12.9 与练习 12.8 的条件相同。绘出下面流函数的等高线图：

$$\Psi=fu_\infty\sqrt{\frac{v_{\text{ius}}x}{u_\infty}}$$

12.10 利用第 12.3.2 节中求解自然对流的方法求 η 的值，条件为 $Pr=0.07$、0.7 和 7.0 且速度 u 为最大。在第 12.3.2 节中，$u=y_2$。当 u 为负数时，调用函数 fminbnd，根据图 12.13 至图 12.15 来选择区间。

[答案：$Pr=0.7$，$u_{\max}=0.278\,4$，$\eta_{\max}=0.953\,5$。]

12.11 根据第 12.3.2 节中求解自然对流的方法，确定在 $Pr=0.07$、0.7 和 7.0 时边界层热量和速度分布区间。在第 12.3.2 节中，边界层温度分布区间 δ_T 被定义为和 η 值相等，$y_4=T^*(\eta=\delta_T)=0.01$，而且边界层速度分布区间 δ_u 被定义为和 η 值相等，y_2

$= u(\eta = \delta_u)/u_{max} = 0.01$，其中 u_{max} 的值是练习 12.4 中的值。

[答案见表 12.11。]

表 12.11　练习 12.11 的答案

Pr	0.07	0.7	7.0
η_{max}	1 267 2	0 953 5	0 691 1
δ_u	13 309	5 800 9	4 876 2
δ_T	13 455	4 537 2	1 825 9

12.12 根据第 12.3.2 节中所描述的空气流过被加热的平板上的自然对流，速度分量可以由下式表示：

$$u = \frac{2v_{\infty}}{x} Gr_x^{1/2} \frac{df}{d\eta}$$

$$v = \frac{v_{\infty}}{x}\left(\frac{Gr_x}{4}\right)^{1/4}\left(\eta \frac{df}{d\eta} - 3f\right)$$

对于静止空气，当温度为 300 K 时计算在 $0 < x < 1$ m，$0 < y < 0.25$ m 范围内的速度分量 $u(x,y)$ 和 $v(x,y)$。将 x 分成 5 段，绘出 u 和 v 的曲线图。参考图 12.12 中定义的 x 和 y。

12.13 在第 12.3.2 节中定义的流函数如下：

$$\Psi = 4vf\left(\frac{Gr_x}{4}\right)^{1\backslash 4}$$

对于温度为 300 K 的气体，绘制出流函数的等高线图，区间为 $0 < x < 1$ m 和 $0 < y < 0.25$ m。参见图 12.12 中 x 和 y 的定义。

12.14 根据第 12.4.1 节的分析过程，改变计算角系数的方程形式，使它适用于本例，在本例中矩形位于两个相互垂直的平面。图 12.24 给出了面积微元与有限矩形分布位置的几何图形。为了简化，只考虑面积微元能看到全部有限矩形这一种情况。也就是说，相互垂直的平面的公共交线不能穿过这个有限矩形。根据表 12.12 给出的数据确定图 12.24 所示表面的角系数，注意在这里 $\cos\theta_1 \neq \cos\theta_2$。

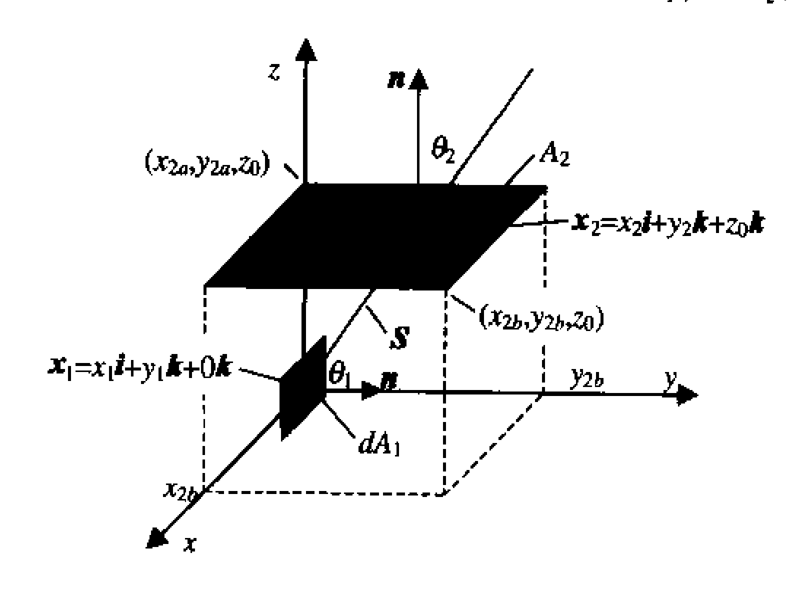


图 12.24　面积微元和有限矩形相互垂直的几何图形

12.15 根据第 12.4.3 节对包壳的分析,把实例中的表面 # 4 分成大小相等的两块后系统有五个面,则应重新计算该例。设表面特性与前例一样,计算每个表面的传热率,并与先前示例的结果相比较。根据第 12.4.3 节给出的表面编号表,表面 4 被分成两个部分,上半部分为 # 4,下半部分为 # 5。这种几何分布的角系数计算结果见表 12.13。这些值是用霍特尔的 crossed-string 方法得到的。结果见表 12.14。

表 12.12　练习 12.14 的参数和答案

参数	组
有限矩形的几何尺寸	
原点 X 坐标	-1
原点 Y 坐标	-1
顶点 X 坐标	0
顶点 Y 坐标	0
微元与垂直平面间的距离	5
角系数计算值 Fd_1-2	0.001 2

表 12.13　练习 12.15 的角系数

i\j	1	2	3	4	5
1	**0**	**0.361 5**	**0.277 0**	**0.095 7**	0.265 8
2	0.216 9	**0**	**0.216 9**	**0.283 1**	0.283 1
3	0.277 0	0.361 5	**0**	**0.265 8**	0.095 7
4	0.114 8	0.566 2	0.319 0	**0**	0
5	0.319 0	0.566 2	0.114 8	0	0

表 12.14　练习 12.15 的答案

表面 #	1	2	3	4	5
Q(kW)	-856 0	8 064	4 451	-2 373	-1 582
T(K)	550	700	650	600	600
q(kW/m^2)	-285 3	1 613	1 484	-949	-633

注意:满足能量守恒,即 Q 的和为零。把这些结果和前面章节中的结果相比较,可以看出在这两类计算中表面 1~3 的传热率(Q)和流量(q)很接近。然而对于表面 # 4,流量随着长度的变化而变化。有意思的是,表面 # 4 和表面 # 5 总的传热率与原先表面 # 4 没有被分开时的传热率非常接近。

12.16 固体球初始温度均匀分布,边界表面具有对流,球的瞬时温度分布由下式给出:[①]

$$\theta(\xi,\tau)=\sum_{n=1}^{\infty}C_n\exp(-\zeta_n^2\tau)\frac{\sin(\zeta_n\xi)}{\zeta_n\xi}$$

其中,

$$\theta(\xi,\tau)=\frac{T(\xi,\tau)-T_\infty}{T(\xi,0)-T_\infty}$$

且 $\tau=\alpha t/a^2$,α 是热扩散系数,a 是球体的半径,t 是时间,$\xi=r/a$,r 是球体径向位置,T_∞ 是周围空气的温度:

---

① 参见 F P Incropera 和 D P DeWitt,出处同上,第 229 页。

$$C_n = \frac{4[\sin\zeta_n - \zeta_n\cos\zeta_n]}{2\zeta_n - \sin 2\zeta_n}$$

ξ_n 是下列方程的正根：

$$1 - \zeta_n\cot\zeta_n = Bi$$

式中 $Bi = ha/k$ 是毕奥特数，h 是传热系数，k 是球体的热导。

当 $0 \leqslant \xi \leqslant 1, 0 \leqslant \tau \leqslant 1.5$ 且 $Bi = 0.5$ 时，绘出 $\theta(\xi,\tau)$随 ξ 和 τ 变化的曲线图。

12.17 三角形肋片上的温度如图 12.25 所示，其上的稳态温度分布由下式确定：

$$(1-\eta)\frac{d^2\theta}{d\eta^2} - \frac{d\theta}{d\eta} - M^2\theta = 0$$

式中：$\eta = x/L$，

$$\theta(\eta) = \frac{T(\eta) - T_\infty}{T_b - T_\infty}$$

$$M^2 = \frac{2hL^2}{kt}\sqrt{1 + \left(\frac{t}{2L}\right)^2}$$

h 为传热系数，k 为热导，设 $t/w << 1$，

设 $\eta = 0$，边界条件 $T(0) = T_b$，即：

$$\theta(0) = 1$$

$\eta = 1$ 时，边界条件为：

$$\left.\frac{d\theta}{d\eta}\right|_{\eta=1} = 0$$

肋片效率 η_f 由下式获得：

$$\eta_f = \frac{-1}{M^2}\left.\frac{d\theta}{d\eta}\right|_{\eta=0}$$

在区间 $0.01 < M^2 < 100$ 内，以 10 的对数坐标等间隔划分，确定肋片效率，并调用函数 semilogx 画出图形。将计算结果与下式的计算结果加以比较：[①]

$$\eta_f = \frac{1}{M}\frac{I_1(2M)}{I_0(2M)}$$

其中，$I_n(x)$为修正的一类 n 阶贝塞尔函数，由函数 besseli 确定。

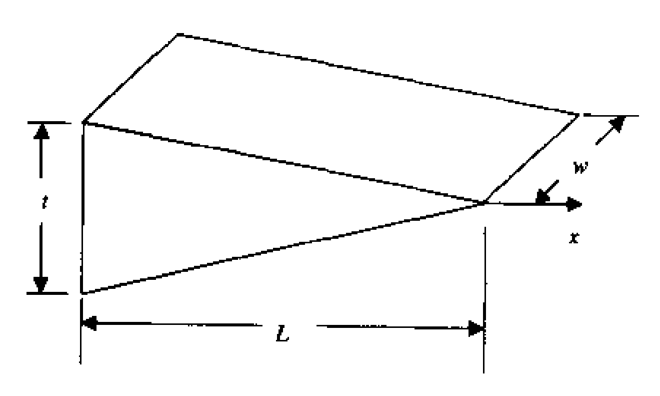


图 12.25 三角形肋片的几何尺寸

① 参见 F P. Incropera 和 D.P. DeWitt，出处同上，第 125。

参考文献

1 Holman J P. Heat Transfer. 7th ed. New York: McGraw Hill, 1990.

2. Incropera F P and DeWitt D P. Fundamentals of Heat and Mass Transfer, 4th ed. New York: John Wiley & Sons, 1996.

3. Kakac S and Liu H. Heat Exchangers: Selection, Rating and Thermal Design. FL: CRC Press, Boca Raton, 1998

4. Krieth F and Bohn M S. Principles of Heat Transfer. 5th ed. New York: West Publishing Co., 1993.

5. Mills A F. Heat Transfer. Boston: Irwin, 1992.

6. Siegel R and Howell J R. Thermal Radiation Heat Transfer. 3rd ed. Washington, DC: Hemisphere Publishing Co., 1992.

7. Suryanarayana N V. Engineering Heat Transfer. New York: West Publishing Co., 1995.

第13章 优　　化

本章介绍如何用 MATLAB 的优化工具箱来解决一系列工程优化问题，给出了典型的示例。

13.1 定义、公式及图解

13.1.1 概述

工程优化是在不同的约束条件下求多变量系统的最优解的过程。在这一定义中“最优”一词意指在一个或多个设计目标中，决策者希望得到的具有最小或最大性能指标的一种设计目标。例如，希望设计一种可靠性最高，重量最轻与成本最低的产品。在工程优化过程中，需要选择描述系统的变量（例如尺寸、形状、材料、工作性能等）。目标是决策者希望得到的尽可能高或尽可能低的量，约束指对系统的某一技术指标有一定的限制。

一般来说，优化问题是在一定约束条件下对一个或多个目标函数进行最小化，通常用下式描述：

$$\min_{x \in D} \{ f_1(x),\ f_2(x)\ ,\cdots, f_m(x) \} \tag{13.1}$$

上式中 $f_i, i = 1, \cdots, m$ 是一标量目标函数，该目标函数将向量变量 $\boldsymbol{x}$ 映射到目标空间中。$\boldsymbol{x}$ 为 n 维决策向量，受约束条件限制分布于可行域 D 中，在可行域的说明中，包含上述优化问题的约束条件，一般可行域由 J 个不等式约束条件或 K 个等式约束条件或两者同时构成，如下所示：

$$D = \{ x:\ g_j(x) \leqslant 0,\ h_k(x) = 0,\ j = 1, \cdots,\ J,\ k = 1, \cdots, K \} \tag{13.2}$$

当目标函数和约束函数均为变量的线性函数时，称之为线性规划；反之，两者至少有一个为非线性函数时，称之为非线性规划。

表 13.1 中列出了优化问题的分类、对应的 MATLAB 求解函数以及本章提供的示例。

表 13.1　优化问题分类、MATLAB 函数和第 13 章的示例

问题分类	MATLAB 函数	示例
线性规划	linprog	13.2 生产计划
非线性规划		
单目标无约束优化		
多变量优化	fminunc	13.1、13.3 两弹簧系统
	fminsearch	
曲线拟合	lsqcurvefit	13.4 应力—应变关系
最小二乘法	lsqnonlin	13.5 应力—应变关系
		13.6 半经验 P-v-T 关系式
单目标约束优化		
单变量优化	fminbnd	13.7 工厂的管道造价问题

（续表）

问题分类	MATLAB 函数	示例
多变量优化	fmincon	13 8 两杆架构问题
		13 9 螺旋形受压弹簧
		13 10 齿轮减速箱
二次项优化	quadprog	13 11 生产计划
半无穷优化	fseminf	13 12 平面两连杆操纵器
多目标优化	fminimax	13 13 振动台问题
	fgoalattain	13 14 生产计划

13.1.2 图解法

两变量优化问题的求解结果可由 MATLAB 的画点功能以图形方式显示。下面给出一个例子。

例 13.1 两弹簧系统

图 13.1 中是两个弹簧组成的系统[①]。图 13.1 中的虚折线是施加载荷前弹簧未变形的情况，图13.1中的实折线是施加一载荷后弹簧变形的情况。在 A 点施加一载荷后系统处于变形状态，经过一定时间后将停留在平衡状态。要研究的是系统的平衡状态，在图 13.1 中实折线具体指出了 A 点的位置(x_1, x_2)。

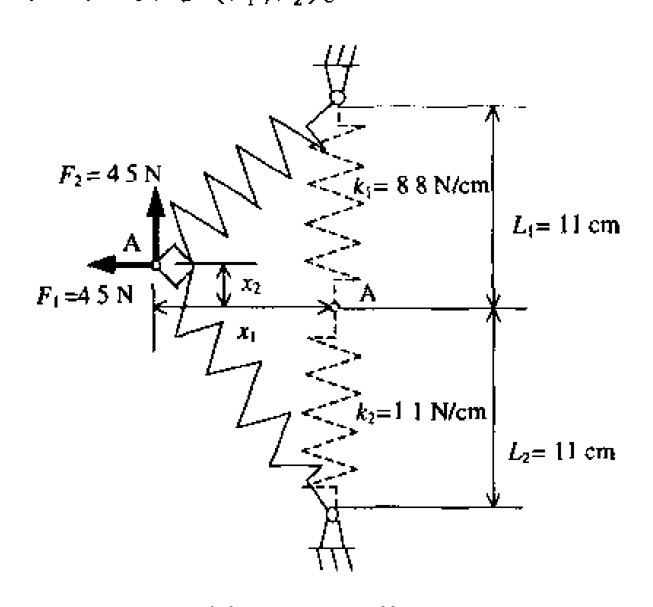


图 13 1 两弹簧系统

为得到系统的平衡状态，首先要算出系统的势能(PE)，然后相对于设计变量 x_1 和 x_2 值求最优值从而求得 A 点位移。势能由弹簧伸长时弹性势能与弹簧在平稳位置处的弹性势能之差计算得出，如式(13.3)中的前两项所示；外力所做的功在式(13.3)中由后两项给出。式中的参数 k_1, k_2, L_1, L_2, P_1, P_2 均为常量，其数值如图 13.1 所示。至此，无约束优

① G Vanderplaats, *Numerical Optimization Techniques for Engineering Design*, McGraw-Hill, New York, 1984, 第 72 ~ 73 页。

化问题的目标函数可以得出：

$$\min_{x_1,x_2} PE(x_1,x_2) = 0.5k_1\left(\sqrt{x_1^2+(L_1-x_2)^2}-L_1\right)^2 + 0.5k_2\left(\sqrt{x_1^2+(L_2+x_2)^2}-L_2\right)^2 - F_1x_1 - F_2x_2 \tag{13.3}$$

目标函数中有两个变量 x_1,x_2，其近似值可由图形估计出。程序清单如下：

```
k1 = 8.8;k2 = 1.1;L1 = 11;L2 = 11;F1 = 4.5;F2 = 4.5;
[x1,x2] = meshgrid(linspace( - 5,15,15),linspace( - 5,15,15));
PE1 = 1/2 * k1 * (sqrt(x1 ^2 + (L1 - x2).^2) - L1).^2;
PE2 = 1/2 * k2 * (sqrt(x1.^2 + (L2 + x2).^2) - L2).^2;
PE = PE1 + PE2 - F1 * x1 - F2 * x2;
subplot(1,2,1);
h = contour(x1,x2,PE,[ - 40:20:20 70:70:490],'k');
clabel(h):
axis([ - 5 15 - 5 15])
subplot(1,2,2);
surfc(x1,x2,PE);
axis([ - 10 15 - 10 15 - 100 500])
```

程序执行结果如图 13.2 所示。图 13.2(a)是 PE 的等高线图，图上标出了每条等高线的数值，从图中可清楚地找到极小值/极大值的分布。图 13.2(b)中上面是 PE 的三维图形，三维图形下面是等高线图，图中也标出了极小值/极大值所在点的近似位置，在该点 PE 目标函数取得极小值/极大值。该例中极小值/极大值所在点的精确位置将在第 13.1.1 节中给出，在第 13.1.1 节中目标函数的优化是由这个无约束优化函数实现的。

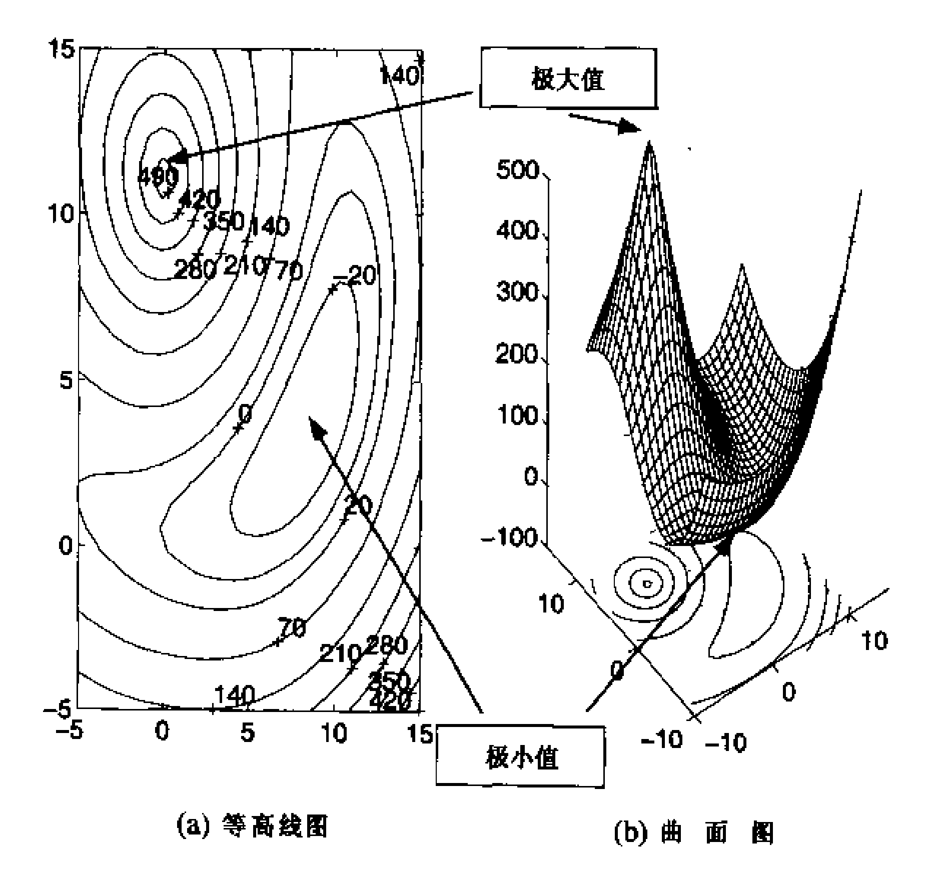


图 13.2　针对图 13.1 给出的两弹簧系统的 PE 函数图

13.2 线性规划

LP(Linear programming,线性规划)是一种优化方法,在优化问题中目标函数和约束函数均为设计变量的线性函数,LP 问题可描述为:

$$
\begin{aligned}
&\min f'x \\
&\text{s.t.(约束条件)}: Ax \leqslant b \\
&A_{eq}x = b_{eq} \qquad (13.4)\\
&L_{bound} \leqslant x \leqslant U_{bound}
\end{aligned}
$$

其中 $\boldsymbol{f}'$, $\boldsymbol{b}$, $\boldsymbol{b}_{eq}$ 为向量,$\boldsymbol{A}$ 和 $\boldsymbol{A}_{eq}$ 为矩阵,x 为一设计向量变量,撇号表示转置,矩阵 $\boldsymbol{A}$ 和向量 $\boldsymbol{b}$ 是线性不等式约束条件的系数,$\boldsymbol{A}_{eq}$ 和 $\boldsymbol{b}_{eq}$ 是等式约束条件的系数。在 MATLAB 中,用于 LP 的求解函数为 linprog,式(13.4)描述的问题可由 linprog 实现求解。linprog 函数的调用格式为:

```
[xopt, fopt] = linprog(f, A, b, Aeq, beq, LBnd, UBnd, x0, options)
```

该函数返回一设计向量变量 x_{opt} 和标量 $f_{(xopt)}$,参数 ***LBnd*** 和 ***UBnd*** 均为向量,分别表示下界 L_{bound} 和上界 U_{bound},$x0$ 为 x 的起始点,*option* 为 optimset 函数中定义的参数的值。

下例为 linprog 函数的用法。

例 13.2 生产计划

有两种液体产品 A 和 B,需要在两个车间加工处理。每件产品 A 在第一车间的处理时间为 1 小时,在第二车间处理时间为 1.25 小时;每件产品 B 在第一车间的处理时间为 1 小时,在第二车间的处理时间为 0.75 小时。每个车间每月有 200 小时的时间可以利用,而且 B 产品的市场需求量最大为 150 件,假定 A 产品和 B 产品的利润每件分别为 \$4 和 \$5,现在需要求出使生产商的利润最大时 A 产品和 B 产品的生产量。

首先,假定 x_1 为 A 产品的生产数量,x_2 为 B 产品的生产数量。则目标函数和约束条件为[①]:

$$
\begin{aligned}
&\min f(x_1, x_2) = -4x_1 - 5x_2 \\
&\text{s.t.}: \\
&\quad g_1: x_1 + x_2 \leqslant 200 \\
&\quad g_2: 1.25x_1 + 0.75x_2 \leqslant 200 \\
&\quad g_3: x_2 \leqslant 150 \\
&\qquad (x_1, x_2) \geqslant 0
\end{aligned}
$$

所以,

$$f' = [-4 \ -5]$$

① 参见 A Osyczka, *Multicriterion Optimization in Engineering with Fortran Programs*, Ellis Horwood Limited, West Sussex, England, 1984 年,第 4 页。

不等式约束表示为：

$$\begin{bmatrix} 1 & 1 \\ 1.25 & 0.75 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leqslant \begin{bmatrix} 200 \\ 200 \\ 150 \end{bmatrix}$$

程序清单为：

```
f=[-4,-5];
A=[1,1;1.25,0.75;0,1];
b=[200,200,150];
LBnd=[0 0];
x=linprog(f,A,b,[],[],LBnd,[])
```

程序的执行结果是 A 产品的生产数量为 $x_1=50$，B 产品的生产数量为 $x_2=150$。

13.3 非线性规划

NLP(Nonlinear programming，非线性规划)指目标函数或约束函数(或两者)为设计变量的非线性函数的一种优化方法。NLP 问题及其相关问题分为两类：无约束优化方法和约束优化方法。

13.3.1 无约束优化

无约束 NLP 方法是确定下列函数式(13.5)表示的无约束变量函数极小值的一种方法：

$$\min_{x} \quad f(x) \tag{13.5}$$

这里，x 为设计向量变量，f 为一标量目标函数。解决上述问题可由两个函数实现，即 fminunc 函数和 fminsearch 函数，其中 fminunc 函数基于梯度搜索法实现，而 fminsearch 函数基于直接搜索法实现。调用 fminunc 的命令如下：

```
[xopt,fopt]=fminunc(UserFunction,x0,options,p1,p2,...)
```

其中 *UserFunction* 是求解目标函数 f 的函数文件名，用单引号括起来，但没有后缀“.m”。如果 *UserFunction* 由 inline 函数生成，则不需要用引号括住。参数 **x0** 表示 x 的起始值，是向量，*options* 为 optimset 定义的参数赋值。*options* 参数带有空矩阵符号[]表示用其默认值。$p1$，$p2$ 等为传递给 *UserFunction* 的参数。

调用 fminsearch 的命令如下：

```
x=fminsearch(UserFunction,x0,options,p1,p2...)
```

其中各个参数的定义与 fminunc 一致。

下面介绍 fminunc 的用法。

例 13.3　两弹簧系统

现在将例 13.1 的两弹簧系统作为无约束优化问题进行数值分析。首先创建设计变量 x_1 和 x_2 的无约束目标函数，程序如下：

```
function PE = SpringEquilibrium(x,k1,k2,L1,L2,F1,F2)
PE1 = 1/2 * k1 * (sqrt(x(1)^2 + (L1 - x(2))^2) - L1)^2;
PE2 = 1/2 * k2 * (sqrt(x(1)^2 + (L2 + x(2))^2) - L2)^2;
PE = PE1 + PE2 - F1 * x(1) - F2 * x(2);
```

然后调用 fminunc 函数实现：

```
x0 = [0.5,5];
k1 = 8.8;k2 = 1 1;L1 = 11;L2 = 11;F1 = 4 5;F2 = 4 5;
options = optimset('MaxIter',600);
[x,f] = fminunc('SpringEquilibrium',x0,options,k1,k2,L1,L2,F1,F2)
```

结果为 $x=[8.4251,3.6331]$，$f=PE=-35.0507$。与图 13.2 中的近似解一致。要获得 PE 函数的极大值，可通过求(1/PE)或(-PE)函数的最优解得到，这时 SpringEquilibrium 函数变为：

```
function PE = SpringEquilibriumMax(x,k1,k2,L1,L2,F1,F2)
PE1 = 1/2 * k1 * (sqrt(x(1)^2 + (L1 - x(2))^2) - L1)^2;
PE2 = 1/2 * k2 * (sqrt(x(1)^2 + (L2 + x(2))^2) - L2)^2;
PE = -(PE1 + PE2 - F1 * x(1) - F2 * x(2));
```

改变后的执行结果为 $x=[0\ 11]$，PE = 549.45，与图 13.2 得到的近似值相比，向量 x 的值是一致的，但是由于画图过程中网格线的缘故，PE 的极大值不一致。此解与估计的起始值及最大迭代次数有关，最大迭代次数增加到 600，才能得到上述结果。

13.3.2 曲线拟合

非线性曲线拟合问题可由求解非线性最小二乘问题的函数 lsqcurvefit 实现。给定输入变量 x_i 的一组值，及其对应的输出变量 y_i 的值，由 lsqcurvefit 函数得到式 $f(x_c,x_i)$的最接近的系数 x_c，即：

$$\min_{x_c}\quad \frac{1}{2}\sum_i[f(x_c,x_i)-y_i]^2 \tag{13.6}$$

MATLAB 用两种优化方法求解式(13.6)描述的问题，默认方法是梯度法与牛顿法的结合，另一种方法为高斯-牛顿方法。函数 lsqcurvefit 的调用格式为：

```
[xopt,res] = lsqcurvefit(UserFunction,x0,xdata,ydata,
                         LBnd,UBnd options,p1,p2,...)
```

其中 $xopt=x_{opt}$ 为 x_c 的最优解，$res=f(x_{opt},x_i)$，*UserFunction* 为求解目标函数的文件名，文件名用单引号括起来，且不包括后缀“.m”。如果 *UserFunction* 由 inline 函数创建，则不需要用引号括起来。参数 $x0$ 为起始值，***xdata*** 和 ***ydata*** 分别为输入输出向量，x_i 和 y_i 以及 *options* 为由 optimset定义的参数赋值。参数 ***LBnd*** 和 ***UBnd*** 为向量，分别表示 x 的下界和上界，即：

$$L_{bound}\leqslant x\leqslant U_{bound}$$

如果不需要 ***LBnd*** 和 ***UBnd*** 参数，其后跟空矩阵符号[]。*options* 参数后跟空矩阵符号[]时，表示用其默认值。$p1,p2$ 等为传递给 *UserFunction* 的参数。

下例给出了 lsqcurvefit 函数的用法。

例 13.4 应力-应变关系

表 13.2 列出了一种塑料材料的一组应力—应变数据，表中 σ 为应力值，单位 ksi。ε 为应变值。假定其关系式具有如下形式：

$$\varepsilon = a + b \ln \sigma \tag{13.7}$$

表 13.2 塑料材料的应力-应变数据

σ	ε
925	0 11
1125	0 16
1620	0 35
2125	0 48
2625	0 61
3125	0 71
3625	0 85

现在要求设计变量 a 和 b，使上式成为表 13.2 中数据的最适合函数。下面的程序中用内嵌函数定义 SigmaEpsilonFit 函数，然后在 lsqcurvefit 函数的第一个参数中引用 SigmaEpsilonFit 函数。程序清单如下所示：

```
sigma = [925,1125,1625,2125,2625,3125,3625];
epsilon = [0.11,0.16,0.35,0 48,0.61,0 71,0.85];
x0 = [0.1,0.1];
SigmaEpsilonFit = inline('x(1) + x(2) * log(sigma)','x','sigma');
[x,resid] = lsqcurvefit(SigmaEpsilonFit,x0,sigma,epsilon)
```

执行结果为：$x(1) = a = -3.581$，$x(2) = b = 0.5344$，残差为 $resid = 0.0064$，代入式(13.7)得到最合适的函数为：

$$\varepsilon = -3.581 + 0.5344 \ln \sigma \tag{13.8}$$

13.3.3 最小二乘法

lsqnonlin 函数可对非线性最小二乘问题求解，此外它也可用于曲线拟合优化中，如例 13.5 所示。但 lsqnonlin 主要用于具有多组输入数据和一组输出数据的问题，如例 13.6 所示。lsqnonlin 函数求出取得极值处的 x 值，问题描述如下：

$$\min_x \sum_i [f_i(x)]^2 \tag{13.9}$$

lsqnonlin 函数的调用方法如下：

```
[xopt,residual] = lsqnonlin(UserFunction,x0,LowBnd,UpBnd,options,p1,p2,...)
```

其中，

$$\text{residual(残差)} = \sum_i [f_i(x_{opt})]^2$$

UserFunction 为求解目标函数的函数文件名，用单引号括起来，但不包括后缀“.m”。如果 *UserFunction* 由函数 inline 创建，则不需用引号括起来。*UserFunction* 创建的目标函数为 $f_i(x)$，$f_i(x)$。参数 **x0** 为起点向量，*LowBnd*，*UpBnd* 为 x 的下界和上界，*options* 为 optimset 定义的参数

赋值，$p1$，$p2$ 等为传递给 *UserFunction* 的参数。如果 *LowBnd* 和 *UpBnd* 后用到空矩阵符[ ]，则表示它们未定义，*options* 后有空矩阵符时表示用默认值。

下面介绍 lsqnonlin 函数的用法。

例 13.5 应力—应变关系

现在用 lsqnonlin 函数求解例 13.4 中的应力—应变问题，首先通过求下式的极小值来确定设计变量 a 和 b：

$$\min_{a,b} \quad \sum_{i=1}^{7}[\varepsilon_i - (a + b\ln\sigma_i)]^2 \tag{13.10}$$

其中 ε_i 和 σ_i 为表 13.2 列出的实验数据。

程序清单如下：

```
sigma = [925,1125,1625,2125,2625,3125,3625];
epsilon = [0.11,0.16,0.35,0.48,0.61,0.71,0.85];
x0 = [0.1,0.1];
SigmaEpsilonLeastSq = inline('epsilon - (x(1) + x(2) * log(sigma))','x','sigma','epsilon');
[x,residual] = lsqnonlin(SigmaEpsilonLeastSq,x0,[],[],[],sigma,epsilon)
```

执行结果为 $x(1) = a = -3.581$，$x(2) = b = 0.5344$，$residual = 0.0064$，与例 13.4 得到的结果相同。

例 13.6 半经验的 P-v-T 关系

众所周知，真实气体的 P-v-T 关系式偏离于理想气体的 P-v-T 关系。理想气体的 P-v-T 关系式为：

$$Pv = RT \tag{13.11}$$

式中 P 是气体的压强，单位 atm；v 为克分子体积，单位 cm^3/g mol；T 为热力学温度，单位 K；R 为气体常数，R = 82.06 atm cm^3/g mol K。在理想气体关系式基础上修正后的半经验关系式为：[①]

$$P = \frac{RT}{v - b} - \frac{a}{v(v + b)\sqrt{T}} \tag{13.12}$$

其中 a，b 的值由实验数据得出。表 13.3 中列出了气体 P-v-T 参数的一些实验数据。通过求下述最小二乘目标函数的极小值可得出设计向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 的值：

$$\min_{\boldsymbol{a},\boldsymbol{b}} \quad \sum_{i=1}^{8}\left[P_i - \frac{RT_i}{v_i - \boldsymbol{b}} + \frac{\boldsymbol{a}}{v_i(v_i + \boldsymbol{b})\sqrt{T_i}}\right]^2 \tag{13.13}$$

式中 P_i、v_i 和 T_i 为表 13.3 中第 i 次的实验数据。如果式(13.12)描述的半经验关系式非常接近实验数据，则在最优点处式(13.13)的目标函数应正好等于 0。但是，由于实验误差及方程的简单化而不能精确地表达非线性，因而在最优点处实际上不会为 0。

程序清单：

```
x0 = [8000,40];R = 82.06;
T = [283,313,375,283,313,375,283,375];
```

① 参见 G V Reklaitis 等，*Engineering optimization*，John Wiley & Sons，New York，1983 年，第 20 ~ 22 页。

```
v=[480,480,576,672,576,672,384,384];
P=[32 7,42 7,47 5,25 7,36.6,38.6,37 6,63 0];
pvt=inline('P-R*T /(v-x(2))+x(1) /(sqrt(T) *v *(v+x(2)))','x','R','T','v','P');
format long e;
options=optimset('MaxFunEvals','600');
[x,residual]=lsqnonlin(pvt,x0,[],[],options,R,T,v,P)
format short
```

执行结果为 $a = x(1) = 5.424 \times 10^7$, $b = x(2) = 3.32$ 和 *residual* = 23.4，由以上值可得出最适合函数为：

$$P = \frac{RT}{v} - \frac{5.425 \times 10^7}{v(v + 3.32)\sqrt{T}} \tag{13.14}$$

表 13.3 气体的 P-v-T 数据

Run i	P(atm)	v,(cm^3/g mol)	T(K)
1	32 7	480	283
2	42 6	480	313
3	44 5	576	375
4	25 7	672	283
5	36 6	576	313
6	38 6	672	375
7	37 6	384	283
8	63 0	384	375

13.4 单目标约束优化

非线性约束优化是对类似式(13.1)和式(13.2)描述的受约束目标函数求最优解的一种方法，当 $m = 1$ 时称为单目标函数。

13.4.1 单变量约束优化

单变量约束优化是对分布于固定区间上的单变量目标函数求最优解的一种方法。

$$\min_{x} f(x) \text{ s.t. } a_1 \leqslant x \leqslant a_2 \tag{13.15}$$

MATLAB 命令为：

```
[xopt,fxopt]=fminbnd(UserFunction,a1,a2,options,p1,p2,...)
```

式中 $xopt = x_{opt}$ 为 x 的最优值，$fxopt = f(x_{opt})$，*UserFunction* 为求解目标函数的函数文件名，用单引号括起来，但没有后缀“.m”。如果 *UserFunction* 由 inline 函数创建，则不需用引号括起来。参数 $a1, a2$ 为 x 所属的区间范围，*options* 为由 optimset 中定义的参数赋值，$p1, p2$ 等为传递给 *UserFunction* 的参数。例 13.7 给出了 fminbnd 函数的用法。

例 13.7　工厂的管道费用问题

若设计一化工厂，管道费用（包括配件及抽灌费用）是需要考虑的一项重要指标。假设设计一条 L 英尺长的管道，抽灌量为 Q g/m，目标是求出管道直径 D（单位 in），使其年度抽灌费用最低。考虑采用标准的碳钢泵，其年度抽灌费用可表示为[①]：

$$f(D) = 0.45L + 0.245LD^{1.5} + 325(hp)^{0.5} + 61.6(hp)^{0.925} + 102 \tag{13.16}$$

其中，

$$hp = 4.4 \times 10^{-8}\frac{LQ^3}{D^5} + 1.92 \times 10^{-9}\frac{LQ^{2.68}}{D^{4.68}} \tag{13.17}$$

下面求管长 1 000 英尺、液体输送速度为 20 g/m 时，管道直径为多少才使其费用最低。首先创建下列函数：

```
function f=PipeLineCost(D,L,Q)
hp=4.4*10^(-8)*L*Q^3/D^5+(1.92*10^(-9)*L*Q^2.68)/(D^4.68);
f=0.45*L+0.245*L*D^1.5+325*hp^0.5+61.6*hp^0.925+102;
```

用下面的语句可求出管道费用最低时的管子直径：

```
L=1000;Q=20;
[D,fD]=fminbnd('PipeLineCost',0.25,6,[],L,Q)
```

执行结果为 $D=1.117$ in，$fD=f(D)=1\,003$ $/年。

13.4.2　多变量约束优化

多变量约束优化是解决非线性多变量约束问题的一种优化方法，由 fmincon 函数实现。约束条件有等式约束和不等式约束，目标函数和约束函数中有一个或多个为非线性函数。非线性多变量约束优化问题可做如下描述：

$$\begin{aligned} &\min_x \quad f(x) \\ &\text{s.t.}: Ax \leqslant b\text{（线性不等式约束）} \\ &\qquad A_{eq}x = b_{eq}\text{（线性等式约束）} \\ &\qquad C(x) \leqslant 0\text{（非线性不等式约束）} \\ &\qquad C_{eq}(x) = 0\text{（非线性等式约束）} \\ &\qquad L_{bound} \leqslant x \leqslant U_{bound} \end{aligned} \tag{13.18}$$

fmincon 函数的调用格式如下：

```
[xopt,fxopt]=fmincon(UserFunction,x0,A,b,Aeq,beq,LBnd,UBnd,
                     'NonLinConstr',options,p1,p2,...)
```

其中 $xopt = x_{opt}$ 为 x 的最优解，$fxopt = f(x_{opt})$ 和 *UserFunction* 为求解目标函数的函数文件名，用单引号括起来，且不包括后缀“.m”。*UserFunction* 中必须给出输出量，其顺序与 fmincon 的帮助文件中定义的顺序一致。如果 *UserFunction* 由 inline 函数创建，则不需用引号括起来。参数 $x0$

① 参见 G V. Reklaitis 等，出处同上，第 66～67 页。

为向量的起始值,矩阵 $\boldsymbol{A}$ 和向量 $\boldsymbol{b}$ 为线性不等式约束条件的系数,矩阵 $\boldsymbol{A}_{eq}$ 和向量 $\boldsymbol{b}$ 为等式约束条件的系数。*LBnd* 和 *UBnd* 分别为 x 的下界和上界,*options* 为 optimset 定义的参数赋值,$p1$,$p2$ 等为传递给 *UserFunction* 和 *NonLinconstr* 的附加参数,*NonLinconstr* 为一函数名,该函数定义了非线性约束条件,其顺序与 fmincon 帮助文件中定义的顺序一致。传递给 *UserFunction* 和 *NonLinconstr* 的参数 $p1$,$p2$ 等必须一致,既使只有一个函数用到时也必须如此。参数 *LowBnd*、*Upbnd*、*options*、*A*、*b*、*Aeq* 和 *beq*,如果没有定义,可用“[ ]”代替。

下面说明 fmincon 的用法。

例 13.8　两杆桁架问题

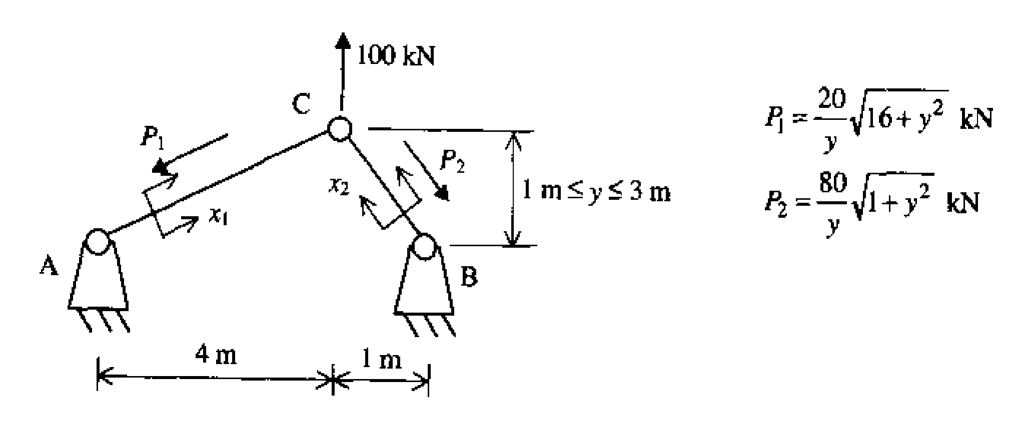


图 13.3　两杆桁架:x_1 和 x_2 为构成横截面的边

如图 13.3 所示的两杆桁架,目标函数为求两根木条 AC 和 BC 构成的架的最小体积。该函数有三个变量,x_1、x_2 和 y。x_1 和 x_2 分别为 AC 和 BC 交叉形成的横断面的两条边,y 为 C 点到水平面的垂直距离。另外还有几个约束条件,两杆的张应力限制为允许值 $\sigma=10^5$ kPa,y 值介于 1 米和 3 米之间,x_1 和 x_2 为非负值。该优化问题可描述如下:[①]

$$\min f_{volume} = x_1\sqrt{16+y^2} + x_2\sqrt{1+y^2}$$

s.t.:

$$\begin{aligned} g_1 &: \frac{20\sqrt{16+y^2}}{yx_1} - \sigma \leqslant 0 \\ g_2 &: \frac{80\sqrt{1+y^2}}{yx_2} - \sigma \leqslant 0 \\ & 1 \leqslant y \leqslant 3 \\ & (x_1, x_2) \geqslant 0 \end{aligned} \tag{13.19}$$

因此,$A=b=A_{eq}=b_{eq}=C_{eq}=0$,$L_{bound}=[1,0,0]$,$U_{bound}=[3,\infty,\infty]$。

程序清单如下:

```
function f = TrussNonLinF(x,sigma)
y = x(1);x1 = x(2);x2 = x(3);
f = x1 * sqrt(16 + y^2) + x2 * sqrt(1 + y^2);
```

① 参见 U Kirsch, *Optimal Structural Design*, McGraw-Hill, New York, 1981 年。

和

```
function[C,Ceq] = TrussNonLinCon(x,sigma)
y = x(1);x1 = x(2);x2 = x(3);
C(1) = 20 * sqrt(16 + y^2) - sigma * y * x1;
C(2) = 80 * sqrt(1 + y^2) - sigma * y * x2;
Ceq = [];
```

程序是：

```
x0 = [1,1,1];sigma = 10^5;
Lbnd = [1 0 0];Ubnd = [3 inf inf];
[x,f] = fmincon('TrussNonLinF',x0,[],[],[],[],Lbnd,Ubnd,'TrussNonLinCon',[],sigma)
```

执行结果为 $x(1)=y=1.952, x(2)=x_1=0.0005, x(3)=x_2=0.0009, f=0.004$。

例 13.9　螺旋形受压弹簧

很多机械装置中都用到了螺旋形受压弹簧，用来施力、提供弹性以及存贮或吸收能量。在设计这种螺旋形受压弹簧时，需要考虑诸如疲劳性、易弯曲性、摇摆、变形之类的一些要求。要满足这些不同的机械要求，可用最优化方法实现求解。描述该问题的公式如下文所述，其中包括一个设计目标函数、两个设计变量、七个约束条件及变量的上界和下界。

设计目标定为求安全系数倒数的极小值，等价于求安全系数的极大值。参考图 13.4，两个设计变量为 c 和 d，其中 $c=D/d$ 为弹簧指数，D 为弹簧的中径，d 为金属丝直径，设计目标为考虑疲劳或易弯曲指标时，求解安全系数倒数的极小值。考虑疲劳指标时，安全系数 SF_f 的倒数由下式确定：

$$\frac{1}{SF_f}=\frac{\tau_a}{S_{ns}}+\frac{\tau_m}{S_{us}} \tag{13.20}$$

式中 τ_a 和 τ_m 分别为剪应力的交变量和平均量，S_{ns} 为弹簧的材料疲劳强度，S_{us} 为其极限强度。考虑易弯曲指标时，安全系数 SF_y 的倒数由下式确定：

$$\frac{1}{SF_y}=\frac{\tau_a+\tau_m}{S_{ys}} \tag{13.21}$$

式中 S_{ys} 为剪切弯曲强度。如果式(13.22)确定的条件满足下式，

$$\frac{\tau_a}{\tau_m}\geqslant\frac{S_{ns}(S_{ys}-S_{us})}{S_{us}(S_{ns}-S_{ys})} \tag{13.22}$$

则目标函数将为式(13.20)给出的考虑疲劳时的安全系数的倒数公式，否则目标函数为式(13.21)给出的考虑易弯曲时的安全系数的倒数公式。剪应力的平均量和交变量分别由下式确定：

$$\begin{aligned}\tau_a&=\frac{8F_acK_w}{\pi d^2}\\ \tau_m&=\frac{8F_mcK_w}{\pi d^2}\end{aligned} \tag{13.23}$$

式中，

$$K_w=\frac{4c-1}{4c+4}+\frac{0.615}{c}$$

$$F_a = (F_U - F_L)/2 \tag{13.24}$$
$$F_m = (F_U + F_L)/2$$

其中 F_U 和 F_L 分别为施加于弹簧轴心的压力的上下限，F_a 和 τ_a 分别为交变力和剪应力，F_m 和 τ_m 分别为平均力和剪应力，K_w 为弹簧的弯曲剪切效应和直接剪切效应的 Wahl 应力系数。

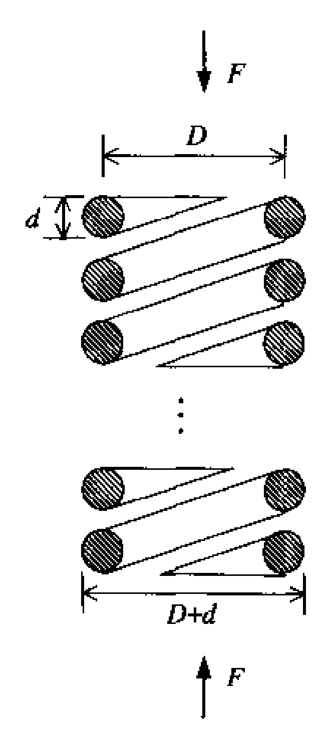


图 13.4　螺旋形受压弹簧

下面给出了螺旋形受压弹簧的全部设计优化公式，考虑疲劳指标时设计目标为式(13.25)表示的形式：

$$\min \quad \frac{1}{SF_f} = \frac{\tau_a}{S_{ns}} + \frac{\tau_m}{S_{us}} \tag{13.25}$$

考虑易弯曲指标时为式(13.26)所示的形式：

$$\min \quad \frac{1}{SF_y} = \frac{\tau_a + \tau_m}{S_{ys}} \tag{13.26}$$

上述目标函数的选择取决于是否满足式(13.22)表示的条件。目标函数的约束条件为：

$$\begin{aligned}
&g_1: K_1 d^2 - c \leqslant 0 && \text{摇摆约束}\\
&g_2: K_2 - c^5 \leqslant 0 && \text{变形约束}\\
&g_3: K_3 c^3 - d \leqslant 0 && \text{弹簧最小圈数约束}\\
&g_4: K_4 d^2 c^{-3} + K_8 d - 1 \leqslant 0 && \text{压缩长度约束}\\
&g_5: K_5 (cd + d) - 1 \leqslant 0 && \text{线圈的最大直径约束}\\
&g_6: c^{-1} + K_6 d^{-1} c^{-1} - 1 \leqslant 0 && \text{线圈的最小直径约束}\\
&g_7: K_7 c^3 - d^2 \leqslant 0 && \text{撞击余量约束}
\end{aligned} \tag{13.27}$$

其中，

$$
\begin{array}{lll}
K_1 = \dfrac{Gf_r\Delta}{112\,800(F_U - F_L)} & K_2 = \dfrac{GF_U(1+A)}{22.3k^2} & K_3 = \dfrac{8kN_{min}}{G} \\
K_4 = \dfrac{G(1+A)}{8kL_m} & K_5 = \dfrac{1}{OD} & K_6 = ID \\
K_7 = \dfrac{0.8(F_U - F_L)}{AG} & K_8 = \dfrac{Q}{L_m} & k = \dfrac{F_U - F_L}{\Delta} \\
S_{ns} = C_1 d^{A_1} \overline{NC}^{B_1} & S_{us} = C_2 d^{A_1} & S_{ys} = C_3 d^{A_1}
\end{array} \tag{13.28}
$$

下面给出上述定义中用到的每个量的假定值：

$A = 0.4$	距离常数(无量纲)
$f_r = 500$ Hz	自然频率的最小允许值
$G = 11.5 \times 10^6$ psi	钢的剪切模量
$ID = 0.75$ in	弹簧内径的最小允许值
$OD = 1.5$ in	弹簧外径的最大允许值
$N_{min} = 3$	最小线圈数
$L_m = 1.25$ in	弹簧承受最大负荷时的最大长度
$Q = 2$	端簧圈数
$\overline{NC} = 10^6$ cycles	失效周数
$\Delta = 0.25$ in	弹簧挠度

假定弹簧材料为钢丝，其 $A_1 = 0.14$, $B_1 = -0.213\,7$, $C_1 = 630\,500$, $C_2 = 160\,000$, $C_3 = 86\,550$，那么弹簧指数 c 和线圈直径 d 的下界和上界分别为：

$$4 \leqslant c \leqslant 20$$

和

$$0.004 \leqslant d \leqslant 0.25$$

在此基础上创建三个函数，第一个为 SpringParameters 函数，用来计算不同的弹簧常数，后两个函数要求这些常数。

```
function[K,Fa,Fm] = SpringParameters(A,FL,FU,G,fr,ID,OD,Lm,Nmin,Q,Delta)
Fa = (FU - FL)/2;
Fm = (FU + FL)/2;
k = (FU - FL)/Delta;
K(1) = G * fr * Delta/(112800 * (FU - FL));
K(2) = G * FU * (1 + A)/(22.3 * k^2);
K(3) = 8 * k * Nmin/G;
K(4) = G * (1 + A)/(8 * k * Lm);
K(5) = 1/OD;
K(6) = ID;
K(7) = 0.8 * (FU - FL)/(A * G);
K(8) = Q/Lm;
```

第二个函数为 SpringNLConstr，用来计算非线性约束条件：

```
function[C,Ceq] = SpringNLConstr(x,K,Fa,Fm,NC,A1,B1,C1,C2,C3)
c = x(1);d = x(2);
```

```
C(1) = K(1) * d^2 - c;
C(2) = K(2) - c^5;
C(3) = K(3) * c^3 - d;
C(4) = K(4) * d^2/c^3 + K(8) * d - 1;
C(5) = K(5) * (c * d + d) - 1;
C(6) = 1/c + K(6)/c/d - 1;
C(7) = K(7) * c^3 - d^2;
Ceq = [];
```

第三个函数为 SpringObjFunct，用来计算目标函数：

```
function f = SpringObjFunct(x,K,Fa,Fm,NC,A1,B1,C1,C2,C3)
c = x(1);d = x(2);
Sns = C1 * d^A1 * NC^B1;
Sus = C2 * d^A1;
Sys = C3 * d^A1;
Kw = (4 * c - 1)/(4 * c + 4) + 0.615/c;
Temp = 8 * c * Kw/(pi * d^2);
TauA = Fa * Temp;
TauM = Fm * Temp;
Ratio = TauA/TauM;
SS = Sns * (Sys - Sus)/(Sus * (Sns - Sys));
if(Ratio - SS) > = 0
 f = TauA/Sns + TauM/Sus;
eles
 f = (TauA + TauM)/Sys;
end
```

则程序为：

```
A = 0.4;FL = 15;FU = 30;G = 11.5e6;
fr = 500;ID = 0.75;OD = 1.5;Lm = 1.25;
NC = 10^6;Nmin = 3;Q = 2;Delta = 0.25;
A1 = -0.14;B1 = -0.2137;
C1 = 630500;C2 = 160000;C3 = 86550;
[K,Fa,Fm] = SpringParameters(A,FL,FU,G,fr,ID,OD,Lm,Nmin,Q,Delta);
x0 = [10,10];
LBnd = [40.004];UBnd = [20 0.25];
[x,f] = fmincon('SpringObjFunct',x0,[],[],[],[],LBnd,UBnd,'SpringNLConstr',[],...
                        K,Fa,Fm,NC,A1,B1,C1,C2,C3)
SafetyFactor = 1/f
```

执行结果为 $x(1) = c = 8.499$，$x(2) = d = 0.100$，安全系数 = 1.84。

例 13.10 齿轮减速箱

如图 13.5 所示，由两个齿轮组成齿轮组，其中一个为主齿轮，另一个为副齿轮。要求设计的齿轮组两轮子的体积及相关的轴体积均最小。该问题中有以下七个设计变量：

x_1 = 齿宽

x_2 = 模

x_3 = 副齿轮的齿数

x_4 = 轴承 1 两端的距离

x_5 = 轴承 2 两端的距离

x_6 = 轴 1 的直径

x_7 = 轴 2 的直径

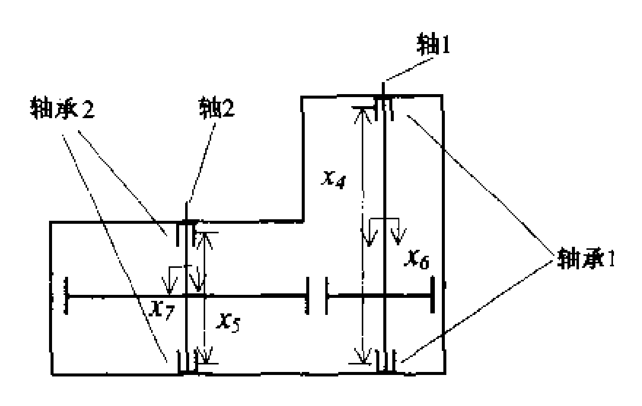


图 13.5 齿轮减速箱

其中每个变量的上下限值为：

$$
\begin{aligned}
2.6 &\leqslant x_1 \leqslant 3.6 \\
0.7 &\leqslant x_2 \leqslant 0.8 \\
17 &\leqslant x_3 \leqslant 28 \\
7.3 &\leqslant x_4 \leqslant 8.3 \\
7.3 &\leqslant x_5 \leqslant 8.3 \\
2.9 &\leqslant x_6 \leqslant 3.9 \\
5.0 &\leqslant x_7 \leqslant 5.5
\end{aligned}
\tag{13.29}
$$

设计目标为求所有轴总体积的最优解，该问题描述如下：

$$
\begin{aligned}
\min f = {} & 0.785\,4x_1x_2^2(3.333\,3x_3^2 + 14.933x_3 - 43.093\,4) - 1.508x_1(x_6^2 + x_7^2) + \\
& 7.477(x_6^3 + x_7^3) + 0.785\,4(x_4x_6^2 + x_5x_7^2)
\end{aligned}
\tag{13.30}
$$

上式中所有变量单位为 cm。其约束条件为：[①]

g_1 : $\dfrac{1}{(x_1x_2^2x_3)} - \dfrac{1}{27} \leqslant 0$　　齿轮的弯曲应力约束

g_2 : $\dfrac{1}{(x_1x_2^2x_3^2)} - \dfrac{1}{397.5} \leqslant 0$　　齿轮的接触应力约束

g_3 : $\dfrac{x_4^3}{(x_2x_3x_6^4)} - \dfrac{1}{1.93} \leqslant 0$　　轴 1 的偏差约束

g_4 : $\dfrac{x_5^3}{(x_2x_3x_7^4)} - \dfrac{1}{1.93} \leqslant 0$　　轴 2 的偏差约束

g_5 : $\dfrac{1}{0.1x_6^3}\sqrt{\left(\dfrac{745x_4}{x_2x_3}\right)^2 + 16.9 \times 10^6} - 1\,100 \leqslant 0$　　轴 1 应力约束

g_6 : $\dfrac{1}{0.1x_7^3}\sqrt{\left(\dfrac{745x_5}{x_2x_3}\right)^2 + 157.5 \times 10^6} - 850 \leqslant 0$　　轴 2 应力约束

① 参见 J Golinski, Optimum Synthesis Problems Solved by Means of Nonlinear Programming and Random Methods, *Journal of Mechanisms*, 5(1970)，第 287～309 页。

$g_7: x_2x_3 - 40 \leqslant 0$ 间隔限制约束

$g_8: 5x_2 - x_1 \leqslant 0$ 间隔限制约束

$g_9: x_1 - 12x_2 \leqslant 0$ 间隔限制约束

$g_{10}: 1.9 - x_4 + 1.5x_6 \leqslant 0$ 轴规格约束

$g_{11}: 1.9 - x_5 + 1.1x_7 \leqslant 0$ 轴规格约束

$g_1 \sim g_7$ 为非线性不等式约束条件，$g_8 \sim g_{11}$ 为线性不等式约束条件。由线性不等式约束条件可得：

$$A = \begin{bmatrix} -1 & 5 & 0 & 0 & 0 & 0 & 0 \\ 1 & -12 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1.5 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1.1 \end{bmatrix}$$

$$b = [0 \quad 0 \quad -1.9 \quad -1.9]'$$

且 $A_{eq} = b_{eq} = C_{eq} = 0$。由式(13.29)得出 $L_{bound} = [2.6, 0.7, 17, 7.3, 7.3, 2.9, 5]$，$U_{bound} = [3.6, 0.8, 28, 8.3, 8.3, 3.9, 5.5]$。

fmincon 用到的两个函数定义如下：

```
function f = GearObjFunct(x)
f = 0.7854 * x(1) * x(2)^2 * (3.3333 * x(3)^2 + 14.9334 * x(3) - 43.0934)...
      - 1.508 * x(1) * (x(6)^2 + x(7)^2) + 7.477 * (x(6)^3 + x(7)^3)...
      + 0.7854 * (x(4) * x(6)^2 + x(5) * x(7)^2);
```

和

```
function[C,Ceq] = GearNonLinConstr(x)
C(1) = 1/(x(1) * x(2)^2 * x(3)) - 1/27;
C(2) = 1/(x(1) * x(2)^2 * x(3)^2) - 1/397.5;
C(3) = x(4)^3/(x(2) * x(3) * x(6)^4) - 1/1.93;
C(4) = x(5)^3/(x(2) * x(3) * x(7)^4) - 1/1.93;
C(5) = sqrt((745 * x(4)/(x(2) * x(3)))^2 + 16.9 * 10^6)/(0.1 * x(6)^3) - 1100;
C(6) = sqrt((745 * x(5)/(x(2) * x(3)))^2 + 157.5 * 10^6)/(0.1 * x(7)^3) - 850;
C(7) = x(2) * x(3) - 40;
Ceq = [];
```

程序为：

```
x0 = [2.6,0.7,17,7.3,7.3,2.9,5];
LBnd = [2.6,0.7,17,7.3,7.3,2.9,5];
UBnd = [3.6,0.8,28,8.3,8.3,3.9,5.5];
A = zeros(4,7);
A(1,1) = -1;A(1,2) = 5;
A(2,1) = 1;A(2,2) = -12;
A(3,4) = -1;A(3,6) = 1.5;
A(4,5) = -1;A(4,7) = 1.1;
b = [0 0 -1.9 -1.9]';
[x,f] = fmincon('GearObjFunct',x0,A,b,[],[],LBnd,UBnd,'GearNonLinConstr')
```

执行结果为：$[x_1, x_2, x_3, x_4, x_5, x_6, x_7] = [3.500, 0.700, 17.000, 7.300, 7.715, 3.350, 5.287]$ 及 $f = 2\,994.3$。

13.4.3 二次规划

二次规划指一类特殊的约束优化问题,其目标函数为二次多项式,约束条件为线性约束,即:

$$\begin{aligned}\min_x f &= 0.5x'Hx + c'x \\ \text{s.t.:}\ Ax &\leqslant b \\ A_{eq}x &= b_{eq} \\ L_{bound} &\leqslant x \leqslant U_{bound}\end{aligned} \tag{13.31}$$

其中 $\boldsymbol{H}$、$\boldsymbol{A}$ 和 $\boldsymbol{A}_{eq}$ 为矩阵,$\boldsymbol{b}$、$\boldsymbol{b}_{eq}$、$\boldsymbol{c}$、$\boldsymbol{x}$、$\boldsymbol{L}_{bound}$ 和 $\boldsymbol{U}_{bound}$ 为列向量,解决该类问题的 MATLAB 函数为:

```
[xopt,fopt] = quadprog(H,f,A,b,Aeq,beq,LBnd,Ubnd,x0,options)
```

其中 $xopt = x_{opt}$ 为 $\boldsymbol{x}$ 的优化值,$fopt = f(x_{opt})$,对称矩阵 $\boldsymbol{H}$ 和向量 $\boldsymbol{c}$ 为二次目标函数 f 的系数,矩阵 $\boldsymbol{A}$ 和向量 $\boldsymbol{b}$ 为非线性不等式约束条件的系数,矩阵 $\boldsymbol{A}_{eq}$ 和向量 $\boldsymbol{b}_{eq}$ 为线性等式约束条件的系数,向量 ***LBnd*** 和 ***Ubnd*** 定义了设计变量 x 的下界和上界。向量 $\boldsymbol{x}0$ 为起点。*Options* 定义 Optimset中参数的值。

下面描述了 quadprog 的用法。

例 13.11 生产计划问题

例 13.2 中讨论的生产计划问题在这里稍微有所改变。现在假定产品 A,B 的利润与其数量之间为一函数关系,对产品 A 来说,单位产品的利润(美元)按下式计算:

$$4 + 2x_1 + 3x_2$$

产品 B 的单位产品利润为:

$$5 + 5x_1 + 4x_2$$

其余条件与例 13.2 中相同,由此得出最优目标函数为:

$$\begin{aligned}f &= -(4 + 2x_1 + 3x_2)x_1 - (5 + 5x_1 + 4x_2)x_2 \\ &= -(4 + 2x_1)x_1 - (5 + 4x_2)x_2 - 8x_1x_2\end{aligned}$$

或

$$f = \frac{1}{2}[x_1 \quad x_2]\begin{bmatrix}-4 & -8 \\ -8 & -8\end{bmatrix}\begin{bmatrix}x_1 \\ x_2\end{bmatrix} + [-4 \quad -5]\begin{bmatrix}x_1 \\ x_2\end{bmatrix} \tag{13.32}$$

程序清单如下:

```
H=[-4,-8;-8,-8];
c=[-4;-5];
A=[1,1;1.25,0.75;0,1];
b=[200;200;150];
LBnd=[0,0];UBnd=[inf,inf];
[xopt,fopt]=quadprog(H,c,A,b,[],[],LBnd,UBnd)
```

执行结果为:$xopt(1) = x_1 = 50$, $xopt(2) = x_2 = 150$, $-fopt = -f(x_{opt}) = 155\,950$。

13.4.4 半无穷约束优化

半无穷约束优化对下述优化问题求解:

$$\begin{aligned}
&\min_x f(x) \\
&\text{s.t.}: Ax \leqslant b \\
&A_{eq}x = b_{eq} \\
&C(x) \leqslant 0 \\
&C_{eq}(x) = 0 \\
&K_1(x,\omega_1) \leqslant 0 \\
&K_2(x,\omega_2) \leqslant 0 \\
&\vdots \\
&K_n(x,\omega_n) \leqslant 0 \\
&\forall(\omega_1,\cdots,\omega_n)
\end{aligned} \tag{13.33}$$

其中 $\boldsymbol{x}$ 为设计向量变量,f 为标量目标函数,$\boldsymbol{A}$ 为表示线性不等式约束的向量,$\boldsymbol{b}$ 为线性不等式约束条件向量,$\boldsymbol{A}_{eq}$ 为表示线性等式约束的向量,$\boldsymbol{b}_{eq}$ 为线性等式约束向量,$\boldsymbol{C}$ 为表示非线性不等式约束条件的向量算子,C_{eq} 为非线性等式约束向量算子。$\boldsymbol{K}_n(x,\omega_n)$为半无穷函数的向量(或矩阵),是 $\boldsymbol{x}$ 向量和 ω_n 的函数,其中自由变量表示一个解的范围,在此范围内求解设计变量 x 的值。$\forall(w_1,...,w_n)$意为自由变量取其相应范围内的所有值,其中 $\forall$ 是"全部"的意思。例如在传热问题中 ω_n 可以表示温度变化范围,在振动问题中可表示频率范围。在变化范围内所有的温度值或频率值必须满足 K_n 约束条件。向量 w_n 的长度最多为 2。

MATLAB 函数为:

```
[xopt,fxopt] = fseminf(UserFunction,x0,n,'SemiConstr',
                A,b,Aeq,Beq,LBnd,UBnd,options,p1,p2,...)
```

其中 $xopt = x_{opt}$ 为 x 的最优解,$fopt = f(x_{opt})$,*UserFunction* 为计算标量函数的函数文件名,用单引号括起来,但没有后缀".m"。如果用 inline 函数创建 *UserFunction*,则变量名不需加引号。$x0$ 参数为起始点,n 为式(13.40)所示的半无穷约束条件数。矩阵 $\boldsymbol{A}$ 和向量 $\boldsymbol{b}$ 为线性不等式约束系数,矩阵 $\boldsymbol{A}_{eq}$ 和向量 $\boldsymbol{b}_{eq}$ 为线性等式约束系数。*LBnd* 和 *Ubnd* 分别定义了设计变量 x 的上界和下届,*options* 设置 optimset 中定义的参数,$p1$,$p2$ 为传递给 *UserFunction* 和 *SemiConstr* 的附加参数。*UserFunction* 和 *SemiConstr* 的 $p1$,$p2$ 等即使有一个函数用到这些值,也必须一致。如果没有定义 *LowBnd*、*Upbnd* 和 *options*,用[]代替。对 $\boldsymbol{A}$、$\boldsymbol{b}$、$\boldsymbol{A}_{eq}$ 和 $\boldsymbol{b}_{eq}$ 也是如此。参数 *SemiConstr* 为一函数名,该函数定义了非线性约束条件,如下所示:

```
[C,Ceq,K1,K2,...,Kn,s] = SemiConstr(x,s,p1,p2,...)
```

其中 $K1,K2,\cdots,Kn$ 分别为估计自由变量在抽样值范围内的半无穷约束条件,两列矩阵 s 各行元素表示相关值的采样间隔,用于计算,即 s 的第 i 行包含了估计的采样间隔。在第一次迭代时,s 置为 NaN,定义采样间隔的初始值。如果 C 和 Ceq 有一项不存在,或两者都不存在,则把它们置为[]。

下面说明函数 fseminf 的用法。

例 13.12 平面两连杆操纵问题

如图 13.6 所示的平面两连杆操纵器,该操纵器能够定位到平面上一点,设计目标为求操

纵器所覆盖的最大工作范围。设计向量表示两连杆的长度，约束条件有下界约束(C_1)与上界约束(C_2)，灵巧性量度系数的上界约束(K_1)，还有设计变量的下界与上界约束 a 和 b。灵巧性(dexterity)① 一词是指操纵器能够很容易地在其工作范围内沿任意方向移动、用力或扭转。雅可比(Jacobian)矩阵的条件系数 κ 用于度量操纵器的灵巧性，其值越接近整数 1 越好。

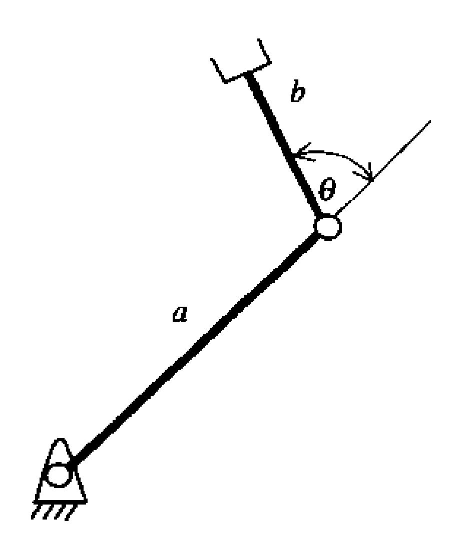


图 13.6 平面两连杆操纵器

下面是对该问题的描述：

$$\begin{aligned} &\min f(a,b) = -\pi[(a+b)^2-(a-b)^2] \\ &\text{s.t.:} \\ &G_1: \quad a/b \geqslant 1.1 \\ &G_2: \quad a/b \leqslant 2 \\ &K_1: \quad \kappa \leqslant 1.26 \\ &\qquad 0.1 \leqslant a \leqslant 2 \\ &\qquad 0.1 \leqslant b \leqslant 2 \\ &\qquad \forall \theta \in [100°,150°] \end{aligned} \tag{13.34}$$

其中条件系数 κ 由下式确定：

$$\kappa = (a^2+2b^2+2ab\cos\theta)/2ab\cos\theta \tag{13.35}$$

半无穷约束条件为满足 $\theta \in [100°,150°]$ 的变化范围，其采样间隔为 5°。首先创建以下函数：

```
function[C,Ceq,K1,s] = TwoLinkConstr(x,s)
a = x(1);b = x(2)
if isnan(s(1,1))
  s = [5 0];
end
theta = (100:s(1,1):150) * pi/180;
```

① 参见 C Gosselin 和 J Angeles, A Global Performance Index for the Kinetic Optimization of Robotic Manipulators, *ASME Journal of Mechanical Design*, 113, (Setp. 1991), 第 222 页。

```
K1 = (a^2 + 2 * b^2 + 2 * (a * b) * cos(theta)) /(2 * (a * b) * sin(theta)) - 1 26;
C(1) = - a/b + 1 1;
C(2) = a/b - 2;
Ceq = [ ];
plot(theta * 180/pi, k1 + 1.26,'k')
hold on
```

估算标量 f 的函数定义如下：

```
function f = TwoLinkObjFunct(x)
a = x(1);b = x(2);
f= - pi * ((a + b)^2 - (a - b)^2);
```

程序为：

```
x0 = [1,1];
LBnd = [.1,.1];UBnd = [2,2];
[x,fopt] = fseminf('TwoLinkObjFunct',x0,1,'TwoLinkConstr',[],[],[],[],LBnd,UBnd)
text(120,1.17,'Initial')
text(113,1.05,'Optimum')
ylabel('Condition number \ kappa')
xlabel(' \ theta (degrees)')
```

程序执行结果形成图 13.7，并给出：$x(1) = a = 2.000$ 及 $x(2) = b = 1.443$，其相应工作范围内的 $-f = 36.2$。图 13.7 中画出了每次迭代过程中条件系数 κ 与 θ 的关系曲线，以显示条件系数从其初始范围到最优范围的变化过程。从图中看出最优线 $\theta = 136°$ 时，条件系数几乎等于 1，该角度即为考虑灵巧性时操作器的最佳形状。

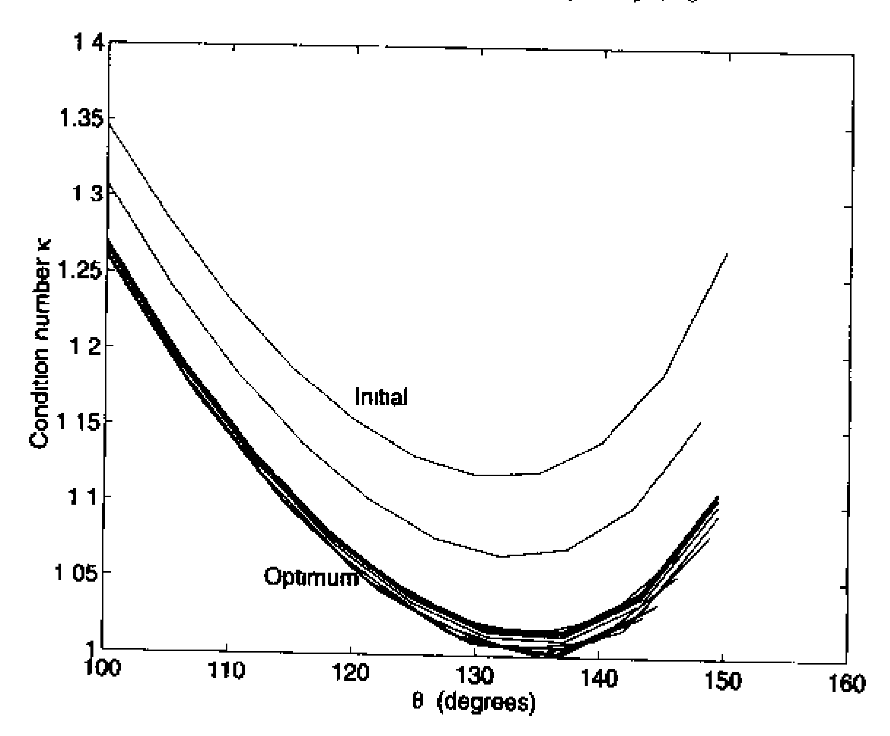


图 13.7 平面两连杆操纵器的条件系数从 a 和 b 的初值变化到最优解的渐近过程，条件系数 κ_1 为 θ 的函数

13.5　多目标优化

多目标优化指设计目标多于一个的优化问题。该类问题中目标相互之间常常是矛盾的，其原因在于问题的内在属性。例如一拉紧构件问题中，两个设计目标分别为求重量的极小值和求应力的极小值。这两个目标相互之间矛盾，即若构件重量减轻，则应力就会增加；反之亦然。在此类优化过程中，求得的点不一定同时满足所有目标，因此多目标问题中的“优化”通常指求得一个解，没有比它更好的解，从而使任何一个目标得到改善。这样一个解被称做非劣点或有效点。多目标优化问题中可能存在许多非劣点，这些点一起形成一条“边界线”，如图13.8为两变量两目标优化问题，图中画出了变量空间的可行域和目标空间的可行域，同时给出了 f_1 和 f_2 都取极小值时的边界线。如图中所示目标空间中的可行域是由变量空间中的可行域曲线映射而来的。边界解集对应于能实现的“最优”解。解集存在一种折中方案，即该解可能使一个目标得到改善而另一目标将变坏。

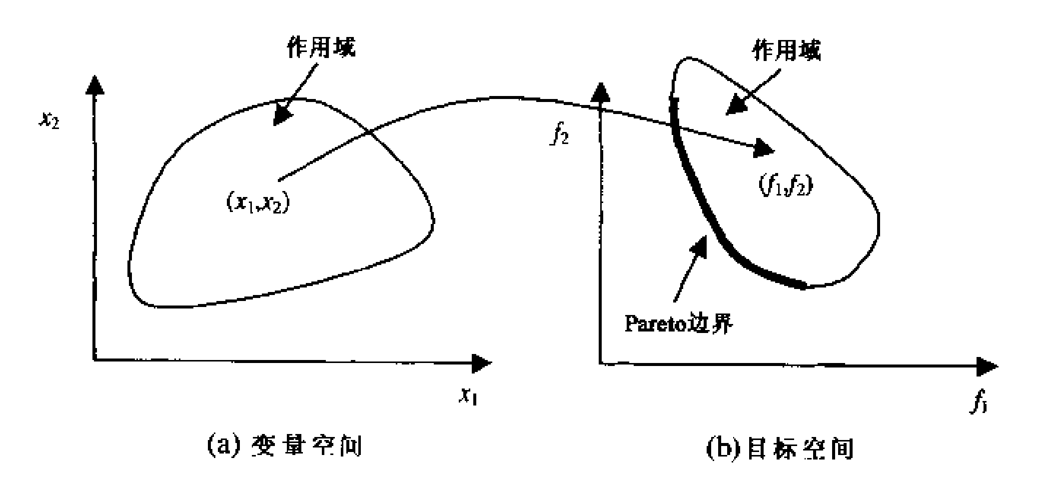


图 13.8　Pareto 边界的作用域

多目标优化问题的最终解将根据决策者的选择从解集中确定。MATLAB 用于解决多目标优化问题的函数有两个：fminimax 和 fgoalattain。

fminimax 方法求解下列问题：

$$\min_{x} \max_{f} \{f_1, f_2, \cdots, f_m\}$$

s.t.：

$$\begin{aligned} & Ax \leqslant b && (\text{线性不等式约束}) \\ & A_{eq}x = b_{eq} && (\text{线性等式约束}) \\ & C(x) \leqslant 0 && (\text{非线性不等式约束}) \\ & C_{eq}(x) = 0 && (\text{非线性等式约束}) \\ & L_{bound} \leqslant x \leqslant U_{bound} && \end{aligned} \tag{13.36}$$

其中 x 为设计向量变量，$f_1, f_2, \cdots, f_m$ 为目标函数，矩阵 A 和向量 b 存放线性不等式约束系数，矩阵 A_{eq} 和向量 b_{eq} 包含线性等式约束系数，C 包含非线性不等式约束，C_{eq} 包含非线性等式约束，L_{bound} 与 U_{bound} 分别为设计量的下界和上界。fminimax 方法使受约束目标函数的最坏值

x 逐次变小。

MATLAB 函数为:

```
[xopt,fxopt] = fminimax(UserFunction,x0,A,b,Aeq,beq,LBnd,UBnd,
                    'NonLinConstr',options,p1,p2,.. )
```

式中 $xopt = x_{opt}$ 为 x 的最优值，$fxopt = f_i(x_{opt})$ 为计算目标函数的函数文件名，用单引号括起来，但不加后缀“.m”。该函数必须以一定顺序生成输出量。当 *UserFunction* 由 inline 创建时，变量名不需用引号括起来。**x0** 为起始值的向量，矩阵 $\boldsymbol{A}$ 和向量 $\boldsymbol{b}$ 为线性不等式约束系数，矩阵 $\boldsymbol{A}_{eq}$ 和向量 $\boldsymbol{b}_{eq}$ 为等式约束系数，***LBnd*** 和 ***UBnd*** 分别为 x 的下界与上界向量，*options* 设置由 optimset定义的参数值，$p1$，$p2$ 等为传递给 *UserFunction* 和 *NonLinConstr* 的附加参数。*NonLinConstr* 为一函数，该函数按照规定顺序定义了一组非线性约束条件，如下所示:

```
[C,Ceq] = NonLinConstr(x,p1,p2,.. )
```

传递给 *UserFunction* 和 *NonLinConstr* 的参数 $p1$，$p2$，…必须保持一致，即使只有一个函数用到这些值或只用到其中一部分也必须如此。如果 *LowBnd*、*Upbnd* 和 *options* 等参数未定义，则用[ ]代替，A、b、Aeq 和 beq 也是如此。

函数 fgoalattain 用于解下述多目标优化问题:

$$
\begin{aligned}
&\max\ \gamma \\
&\text{s.t.:} \\
&\quad f_i(x) - \omega_i\gamma \leqslant (goal)_i \qquad i = 1,\cdots,m \\
&\quad Ax \leqslant b \qquad \text{(线性不等式约束)} \\
&\quad A_{eq}x = b_{eq} \qquad \text{(线性等式约束)} \\
&\quad C(x) \leqslant 0 \qquad \text{(非线性不等式约束)} \\
&\quad C_{eq}(x) = 0 \qquad \text{(非线性等式约束)} \\
&\quad L_{bound} \leqslant x \leqslant U_{bound}
\end{aligned}
\tag{13.37}
$$

其中 γ 为可正可负的标量变量，f_i 为第 i 个目标函数，w_i 和 $(goal)_i$ 分别为第 i 个目标函数的加权系数和目标。加权系数控制未达到目标或超过目标的相对程度，$w_i\gamma$ 在公式中为一弱化量。例如设所有加权系数均等于其初始目标，意味着期望未达到目标的百分比与超越目标的百分比相同。

MATLAB 函数为:

```
x = fgoalattain(UserFunction,x0,Goal,Weight,A,b,Aeq,beq,LBnd,UBnd,
                    'NonLinConstr',options,p1,p2,...)
```

其中向量 ***Weighf*** 包含元素 w_i，向量 ***Goal*** 包含元素 $(goal)_i$，其余参数与 fminimax 中定义的相同。

下面的两个例子给出了 fminimax 和 fgoalattain 的应用。

例 13.13 振动台

在图 13.9 所示的系统中，安装有发动机的一梁型平台由三层材料构成，假定梁两端有简单的支撑物，由发动机引起的振动能够传递给梁。

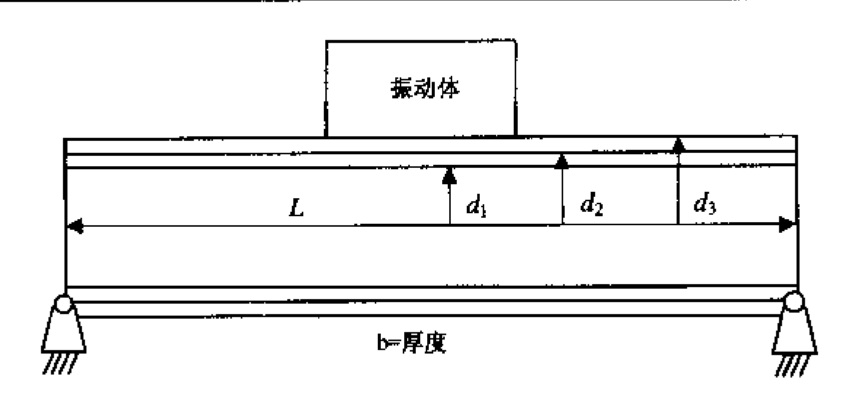


图 13.9 由多层材料构成的简单支撑振动台

该系统的设计目标为：

1. 对梁的基本自然频率的负值求极小值，用 f_1 表示。
2. 对梁所用材料的费用求极小值，用 f_2 表示。

约束条件包括：梁 g_1 的质量上限，第二层梁 g_2 和第三层梁 g_3 厚度的上限，还有设计变量的上下限。五个设计变量为梁的长度 L，宽度 b，到第一层 d_1 和第二层 d_2 接触面到梁的中心轴距离，以及到梁顶的距离 d_3。表 13.4 列出了三层材料中每层的质量密度 ρ，弹性模量 E 及单位体积 c 的费用。问题描述如下：

$$\min \quad f_1(d_1, d_2, d_3, b, L) = -(\pi/2L^2)\sqrt{EI/\mu}$$
$$\min \quad f_2(d_1, d_2, d_3, b) = 2b(c_1 d_1 + c_2(d_2 - d_1) + c_3(d_3 - d_2))$$

其中，

$$EI = (2b/3)(E_1 d_1^3 + E_2(d_2^3 - d_1^3) + E_3(d_3^3 - d_2^3))$$
$$\mu = 2b(\rho_1 d_1 + \rho_2(d_2 - d_1) + \rho_3(d_3 - d_2))$$

约束条件为：

$$g_1 : \mu L - 2\,800 \leqslant 0 \qquad \text{梁的质量}$$
$$g_2 : d_2 - d_1 - 0.15 \leqslant 0 \qquad \text{层厚}$$
$$g_3 : d_3 - d_2 - 0.01 \leqslant 0 \qquad \text{层厚}$$
$$0.05 \leqslant d_1 \leqslant 0.5$$
$$0.2 \leqslant d_2 \leqslant 0.5$$
$$0.2 \leqslant d_3 \leqslant 0.6$$
$$0.35 \leqslant b \leqslant 0.5$$
$$3 \leqslant L \leqslant 6$$

表 13.4 振动台的材料性能指标及费用

层 i	ρ_i(Kg/m³)	E_i(N/m²)	c_i($/体积)
1	100	1.6×10^9	500
2	2 770	70×10^9	1 500
3	7 780	200×10^9	800

由此看出 g_1 为非线性不等式约束，g_2 和 g_3 为线性不等式约束，因此：

$$A = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \end{bmatrix}$$

$$b = [0.15 \quad 0.01]'$$

因为没有线性和非线性等式约束，所以 $C_{eq} = A_{eq} = b_{eq} = 0$。

为保持与计算函数有相同的数量级，设计目标换算为下式：

$$scaled_value = \frac{raw - good}{bad - good} \tag{13.38}$$

式中 *raw* 指函数实际值(换算前)，*good* 指函数的理想值，*bad* 指非理想值。从式(13.38)可以看出，当 *raw* 与 *good* 相等时，$scaled_value = 0$；当 *raw* 与 *bad* 相等时，$scaled_value = 1$。

下面创建三个函数，其中第一个函数用于计算 EI 和 μ，并为后两个函数所用。

```
function[EI,mu] = BeamProperties(x,E,Rho,c)
EI = (2 * x(4)/3) * (E(1) * x(1)^3 + E(2) * (x(2)^3 - x(1)^3) + E(3) * (x(3)^3 - x(2)^3));
mu = 2 * x(4) * (Rho(1) * x(1) + Rho(2) * (x(2) - x(1)) + Rho(3) * (x(3) - x(2)));
```

其中 $[x_1, x_2, x_3, x_4, x_5] = [d_1, d_2, d_3, b, L]$。第二个函数计算非线性不等式约束：

```
function[C,Ceq] = VibPlatNLConstr(x,E,Rho,c,good,bad)
[EI,mu] = BeamProperties(x,E,Rho,c);
C(1) = mu * x(5) - 2800;
Ceq = [];
```

第三个函数计算目标函数：

```
function f = VibPlatformObj(x,E,Rho,c,good,bad)
[EI,mu] = BeamProperties(x,E,Rho,c);
f1 = pi/(2 * x(5)^2) * sqrt(EI/mu);
f(1) = (f1 - good(1)/(bad(1) - good(1));
f2 = 2 * x(4) * (c(1) * x(1) + c(2) * (x(2) - x(1)) + c(3) * (x(3) - x(2)));
f(2) = (f2 - good(2))/(bad(2) - good(2));
```

程序为：

```
x0 = [0.3,0.35,0.4,5,0.4];
LBnd = [0.05,0.2,0.2,0.35,3]
UBnd = [0.5,0.5,0.6,0.5,6]
E = [1.6,70,200] * 10^9;
Rho = [100,2770,7780];
c = [500,1500,800];
good = [500,100];
A = [-1 1 0 0 0;0 -1 1 0 0];
b = [0.15 0.01]';
for k = 1:5
  bad = [100 + k * 10 500 - k * 50]
  [xopt,fxopt] = fminimax('VibPlatformObj',x0,A,b,[],[],LBnd,UBnd,...
                'VibPlatNLConstr',[],E,Rho,c,good,bad);
  for m = 1:2
    ff(m) = fxopt(m) * (bad(m) - good(m)) + good(m);
  end
  f1(k) = ff(1);
```

```
    f2(k) = ff(2);
end
[f2sort indxf2] = sort(f2);
f1sort = f1(indxf2);
plot( - f1sort,f2sort,'ko-');
xlabel('Negative frequency(Hz)');
ylabel('Cost( $ )');
title('Trade-off(Pareto) solutions');
```

上述程序执行结果为图 13.10 所示的边界线。

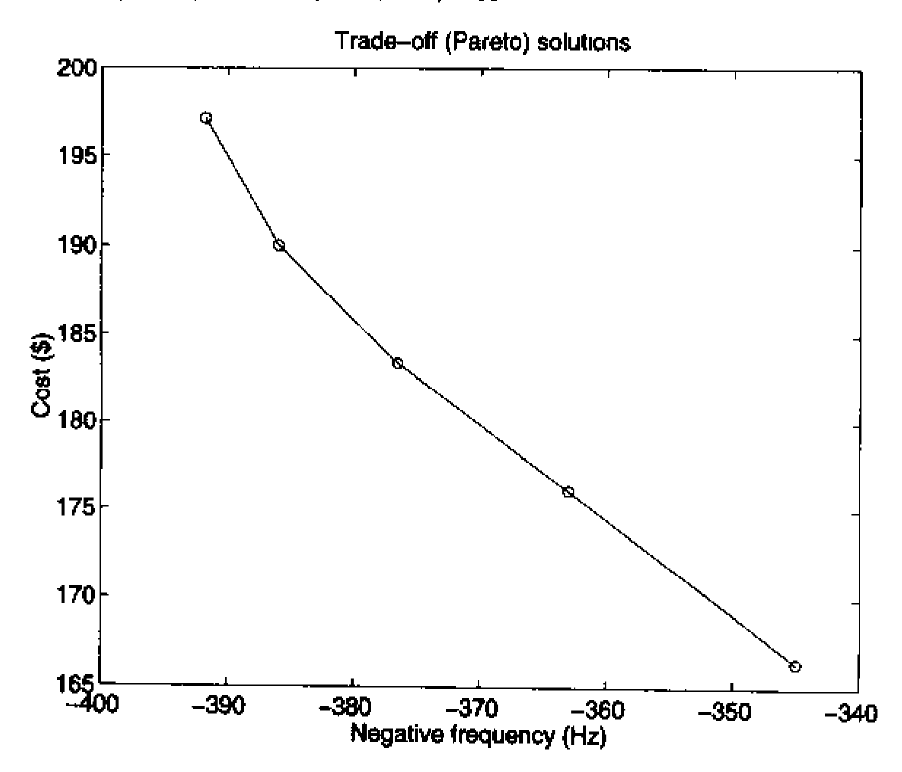


图 13.10 振动台的 Pareto 边界线

例 13.14 生产计划问题

再次考虑例 13.2 中的生产计划问题。现在介绍第二种设计目标,即对产品 A 的生产件数求极大值(或对其负值求极小值)。下面是其目标函数和约束条件:

$$\min f(x_1,x_2) = -4x_1 - 5x_2$$
$$f(x_1) = -x_1$$

s.t.:

$$g_1: x_1 + x_2 \leqslant 200$$
$$g_2: 1.25x_1 + 0.75x_2 \leqslant 200$$
$$g_3: x_2 \leqslant 150$$
$$(x_1,x_2) \geqslant 0$$

因此,

$$A = \begin{bmatrix} 1 & 1 \\ 1.25 & 0.75 \\ 0 & 1 \end{bmatrix}$$

$$b = [200 \quad 200 \quad 150]'$$

可以得到其下限 $L_{bound}=[0\ 0]$和上限 $U_{bound}=[\infty\ \ \infty]$。这里没有等式约束和非线性约束条件，所以 $C_{eq}=A_{eq}=b_{eq}=C=0$。

程序代码如下：

```
A=[1 1;1.25 0 75;0 1];
b=[200 200 150]';
goal=[-950 -50];x0=[50 50];
LBnd=[0 0 ];UBnd=[inf inf];
Weight=abs(goal);
options=optimset('GoalsExactAchieve','2');
ProdPlanObj=inline('[-4*x(1)-5*x(2),-x(1)]','x');
[x,fevall]=fgoalattain(ProdPlanObj,x0,goal,Weight,A,b,[],[],...
                        LBnd,UBnd,[],options)
```

其中参数 *GoalsExactAchieve* 值设为2(独立变量数)，这是一种正好达到目标的算法，既不过也不欠。如果用默认值，则解为 $x0$ 的一个函数，程序执行结果得到一个解$[x_1,x_2]=[50,150]$及 *fevall* $=[-950\ \ -50]$。

练习

13.1 如图13.11所示机械装置，其中棒 *OA* 和 *AC* 长度均为 $L_1=0.5$ m，*CB* 和 *BD* 长均为 $L_2=0.3$ m，*O* 点固定，*C* 点和 *D* 点分别钉在两滑块中，并且可沿水平方向无摩擦移动。该装置受三个外力作用，其中在 *A* 点和 *B* 点的垂直力分别为 $P1=3$ kN 和 $P2=1$ kN。*D* 点有一水平方向力 $P=3$ kN。假定未施力之前该装置中的四根棒沿水平线调成一直线，忽略该装置的重量，则该装置在施加外力时的平衡状态可通过求其势能(*PE*)函数的极小值得到：

$$\min PE=-P_1L_1\sin\alpha-P_2L_2\sin\beta+P[L_1(1-\cos\alpha)+L_2(1-\cos\beta)]$$

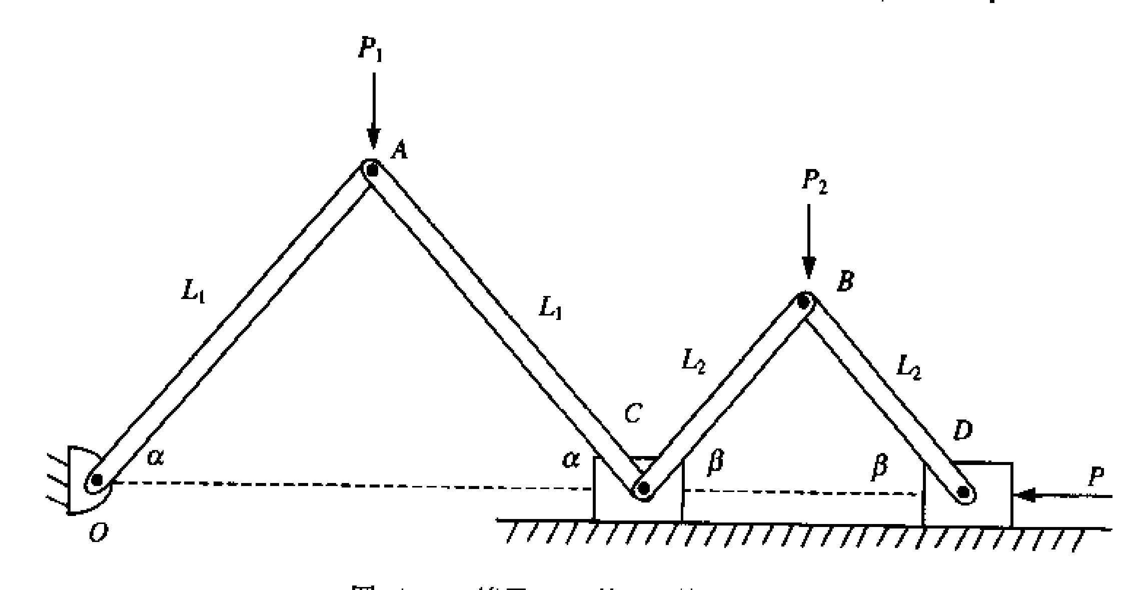


图13 11 练习13.1的两对棒机械装置

(a) 给出 *PE* 的等高线图α 和曲线图β，并从中求得的最优弧度值。

(b) 用函数 fminunc 计算并证实上述结果。

[答案：$\alpha = 0.785$ rad，$\beta = 0.322$ rad。]

13.2 如图 13.12 所示的管道[1]，要求设计一固定横剖面的管道，使其水流速度最大。设计变量为高度 h，底部宽 c，边角 θ。已知流速与润湿周长 p 的倒数成比例关系，而 p 由下式确定：

$$p = c + (2h/\sin\theta)$$

横剖面面积 A 为：

$$A = ch + h^2\cot\theta$$

已知 $A = 100\ \text{ft}^2$，求：

(a) 给出无约束优化问题的定义，以求得流速的最大值，其中流速为设计变量的函数。

(b) 绘出图 13.13。

(c) 用 fminunc 函数验证由图形得到的解的正确性，给定初始点 $(h,\theta) = (1,1)$。

[答案：$[h,\theta] = [7.59, 1.01$ rad。]

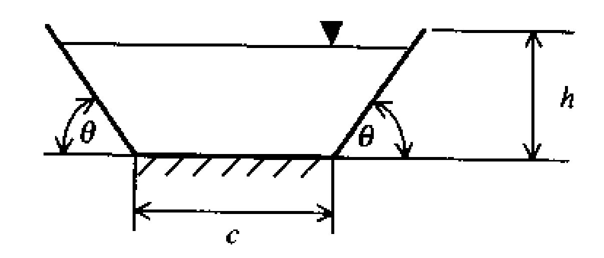


图 13.12 练习 13.2 的水管道问题

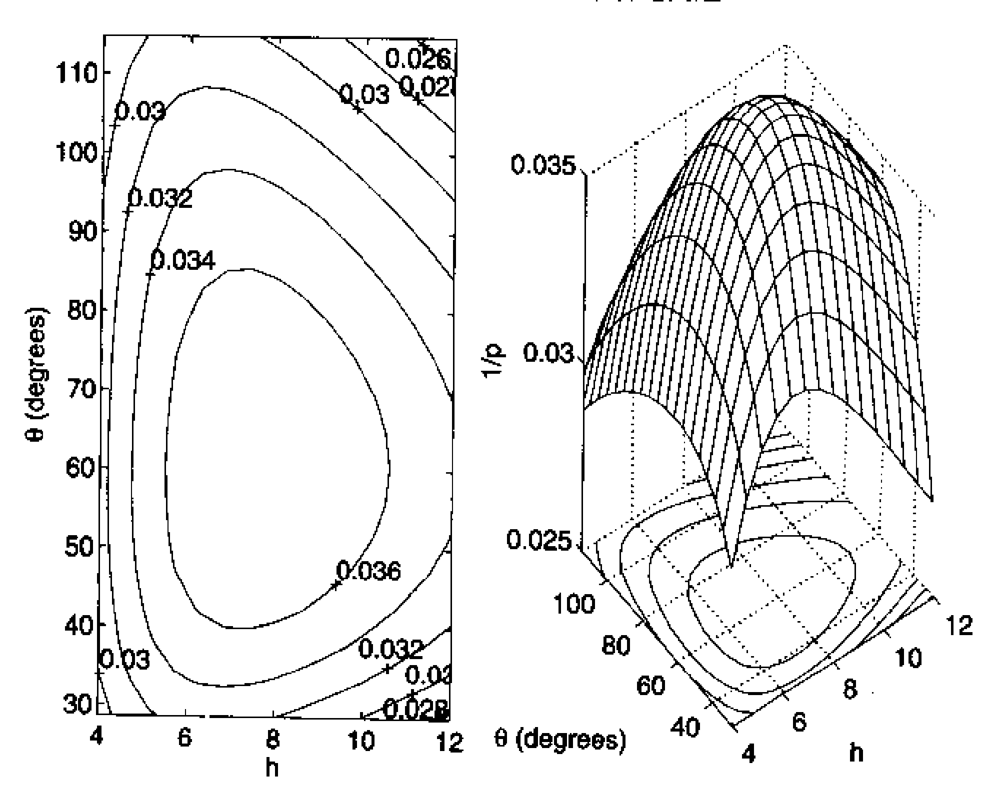


图 13.13 练习 13.2 的等高线和曲面线图

---

[1] 参见 P.Y. Papalambros 和 D.J. Whild, *Principles of Optimal Design: Modeling and Computation*, Cambridge University Press, New York, 1988 年，第 151 页。

13.3 对一工件进行机械加工所需平均时间由下式确定：[1]

$$T = t_m + \frac{t_m t_c}{T_l} + t_{aux}$$

式中 t_m 为切削时间，T_l 为工具寿命，t_c 为换工具时间，t_{aux} 为附加时间。给定经验常数 $t_c = 7$，$t_{aux} = 3$，在一车削操作中，切削时间由下式确定：

$$t_m = \frac{\pi DL}{1\,000 Vf}$$

式中已加工面直径为 D = 100 mm，需加工的工件长为 L = 500 mm，V 和 f 分别为表面切削速度和切削工具的进刀速度。切削速度、进刀速度和切削深度 d 对工具寿命的影响程度由泰勒展开式求出，如下所述：

$$VT_l^n f^a d^b = K_t$$

式中 $n = 0.17$，$a = 0.77$，$b = 0.37$，$K_t = 200$，V 和 f 为设计变量，则此优化问题可描述如下：

$$\min \ T(V,f) \qquad \text{s.t.}: f \leqslant 2$$

给定 $d = 0.3$，求：

(a) 绘出其等高线图。

(b) 用 fmincon 函数进行数值分析以求得最优解。

[答案：$T = 3.94$，$V = 100$，$f = 2$。]

13.4 图 13.14 中两个无摩擦拖车 A 和 B 由三个弹簧相连，弹簧弹力与其伸长量之间为线性关系，弹簧的倔强系数分别为 $k_1 = 5$ N/m，$K_2 = 10$ N/m 和 $K_3 = 8$ N/m[2]。当外力 P 为 0 时，弹簧处于自然平衡位置，给出热能函数如下，试画出等高线图和曲面图，并计算 $P = 100$ N 时，位移 x_1 与 x_2 的解。

$$\min \quad PE = 0.5k_2x_1^2 + 0.5k_3(x_2 - x_1)^2 + 0.5k_1x_2^2 - Px_2$$

[答案：$PE = 530$，$x_1 = 4.5$，$x_2 = 13.5$。]

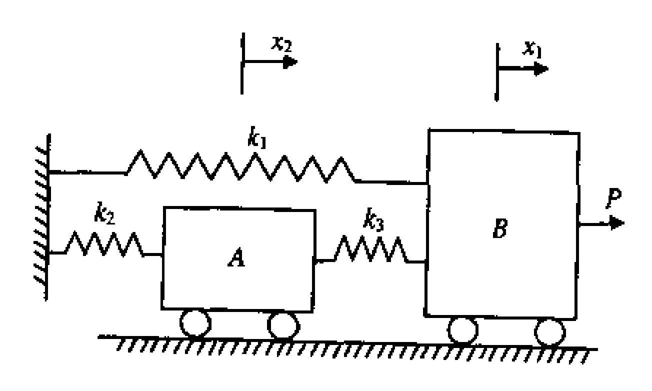


图 13.14 练习 13.4 中的弹簧—小车系统

13.5 如图 13.15 所示[3]，有三个小车（重物）用弹簧连接，开始处于未受力平衡状态，当分别施加外力 P_1，P_2，P_3 时，小车状态将改变，三小车从其初始平衡位置（$x_i = 0$，i

---

① 参见 D A Stephenson 和 J S Agapiou, *Metal Cutting Theory and Practice*, Marcel Dekker, New York, 1997 年。

② 参见 S S Rao, *Engineering Optimization, Theory and Practice*, 第二版, John Wiley & Sons, New York, 1996 年。

③ 参见 S S Rao, 出处同上, 第 420 页。

值取 1,2,3)处开始振动,其位移可通过求系统的势能(*PE*)的极小值求得:

$$PE = 0.5X'KX - X'P$$

其中,

$$K = \begin{bmatrix} k_1 + k_2 + k_4 & -k_3 & -k_4 \\ -k_3 & k_2 + k_3 + k_5 & -k_5 \\ -k_4 & -k_5 & k_4 + k_5 + k_6 \end{bmatrix}$$

$$P = [P_1 \quad P_2 \quad P_3]'$$

$$X = [x_1 \quad x_2 \quad x_3]'$$

输入数据为:

$k_1 = 4\,500$ N/m	$k_4 = 2\,250$ N/m	$P_1 = 1\,100$ N
$k_2 = 1\,650$ N/m	$k_5 = 550$ N/m	$P_2 = 1\,800$ N
$k_3 = 1\,100$ N/m	$k_6 = 9\,300$ N/m	$P_3 = 3\,300$ N

试用 fminunc 函数求出三小车的平衡位置。

[答案:$[x_1, x_2, x_3] = [0.322, 0.714, 0.365。]$]

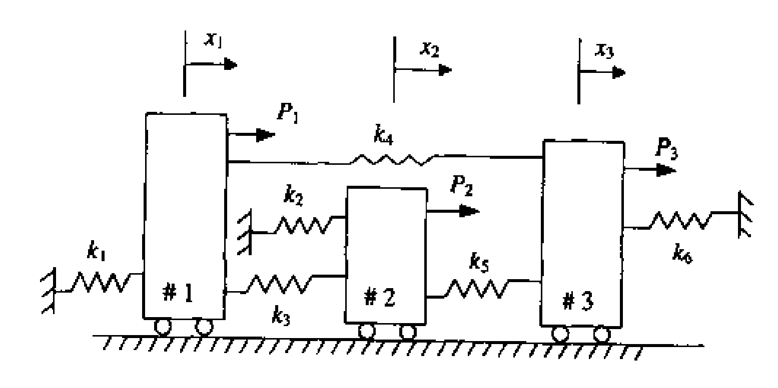


图 13.15　练习 13.5 的弹簧—小车系统

13.6　如图 13.16 所示的弹簧—重物系统,未加重物时弹簧处于不变形的位置,当在每段弹簧之间挂上重物时弹簧将变形①。第 i 根弹簧的稳定性用 k_i 表示,由下式得出:

$$k_i = 450 + 225(N/3 - i)^2 \qquad \text{N/m} \qquad i = 1,\cdots,6$$

其中 $N = 5$,是所挂重物的个数,W_j 由下式确定:

$$W_j = 60j \qquad N \quad j = 1,\cdots,5$$

挂重物前每段弹簧长为 $L_i = 7.5$ m,弹簧连接点(从 2 到 6)的坐标用 10 个设计变量表示。可通过求下述 *PE* 函数的极小值来获得其平衡位置:

$$PE = 0.5\sum_{i=1}^{6} K_i \Delta L_i^2 + \sum_{j=1}^{5} W_j Y_j$$

其中,

$$\Delta L_i = \sqrt{(X_{i+1} - X_i)^2 + (Y_{i+1} - Y_i)^2} - L_i$$

要求用 fminunc 求出平衡位置,即图 13.16 中弹簧变形后的各连接点坐标。

① 参见 G N Vanderplaats,出处同上,第 94 页。

[答案：$X_2=7.92,X_3=16.25,X_4=24.36,X_5=32.22,X_6=39.30,Y_2=-3.86,Y_3=-7.15,Y_4=-8.84,Y_5=-8.29,Y_6=-5.16$。]

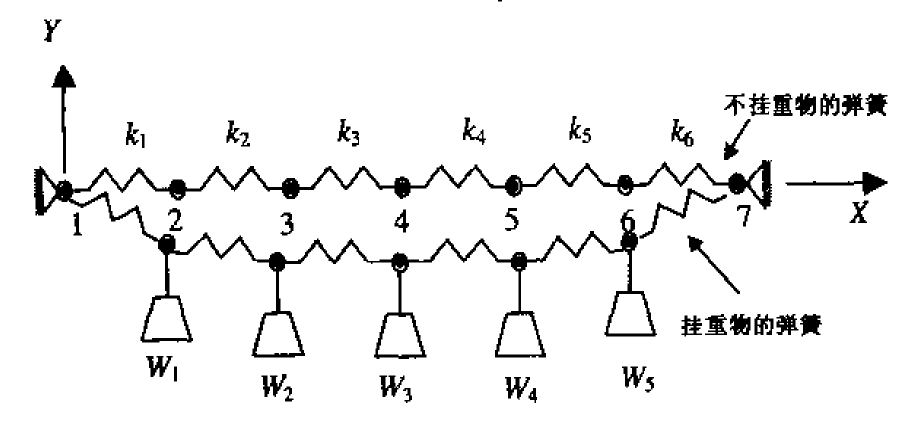


图 13.16 练习 13.6 中的弹簧—重物系统的初始位置和变形后位置

13.7 如图 13.17 所示，已知管柱的弯曲临界载荷 P 符合下述方程式，其中 a,b 为未知常数：

$$P=\frac{\pi^a E R^b t}{4l^2}$$

式中 E 为弹性模量，R 为平均半径，t 为厚度，l 为柱长。假定导致弯曲的精确关系式未知，常数将由实验数据经曲线拟合得到。为此进行了实验，实验所用柱体半径不同，弹性模量 $E=250$ ksi，长度 $l=5$ in，厚度 $t=1$ in，给柱施加屈曲载荷。表 13.5列出了不同 R 值对应的屈曲载荷 P。要求用 lsqcurvefit 函数求 a 和 b，重新画出图 13.18，并在图中给出实验数据和曲线拟合数据。

[答案：$a=-3.318\,1,b=3.159\,9$。]

表 13.5 练习 13.7 中的半径 R 和载荷 P 的实验数据

实验次数	R(in)	P(lb)
1	1.1	86 6
2	1.7	120 5
3	2 1	520.88
4	2.9	1 758
5	3.9	4 098

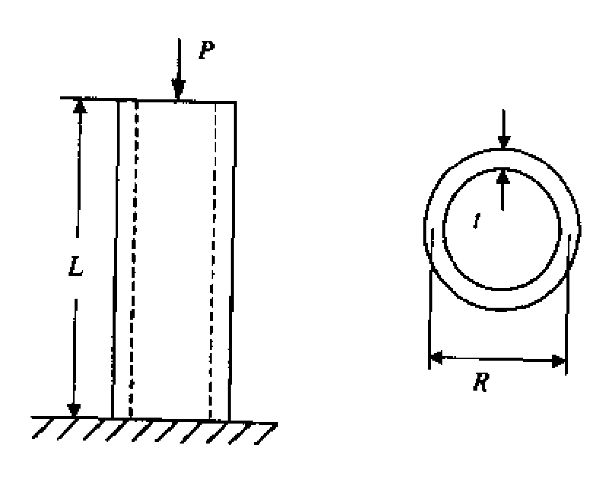


图 13.17 练习 13 7 中的柱体几何形状

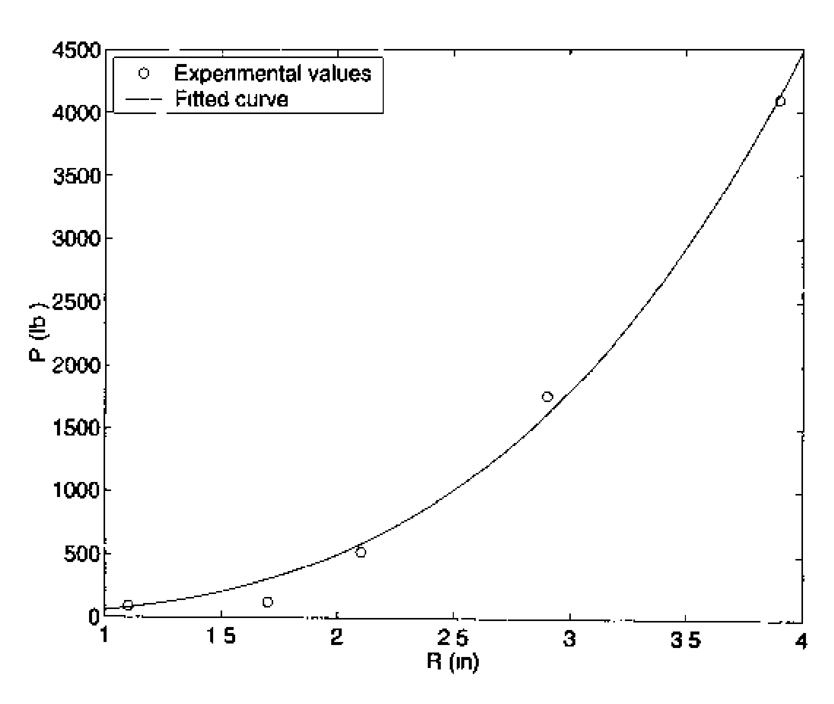


图 13.18　练习 13.7 中管状柱体问题的结果曲线

13.8 假设一种药品中某一成分含量应保持在药品总量的 0.5%，而这种成分随时间的推移会减少。药品上市前八个星期内，存在着下降到 0.5% 的可能，另外还会有其他不可控因素的出现，所以用理论方法预测其今后的下降趋势是不可行的。要辅助管理者做出是否对库存过期药品废弃或更换的决定，应定期分析药品以测量药品成分的含量。表 13.6 给出其测量值，用下述的非线性模型可说明表中数据的变化规律。现要求用 lsqcurvefit 函数估算出非线性模型中的参数 a 和 b 值。
[答案：$[a,b]=[0.392,0.139]$。]

表 13.6　练习 13.8 的数据表

自生产之日起的时间长度(周数) t	成分含量 Y	自生产之日起的时间长度(周数) t	成分含量 Y
7	0 488	25	0 405
9	0 473	27	0 403
11	0 448	29	0 391
13	0.435	31	0 403
15	0 431	33	0 398
17	0 453	35	0 393
19	0 421	37	0 398
21	0 405	39	0 388
23	0.405	41	0 388

13.9 观察停车处的车辆，车停下来然后又加速行驶越过五座房屋。假设这五座房屋与停车处的距离已知，从车开始加速行驶时刻算起，到它驶过每座房屋时记下其行驶时间，表 13.7 中列出了这些数据。车辆加速行驶问题是时间的函数，其形式为：

$$a(t) = Ct^2 + Dt + a_0$$

式中 t 为行驶时间(s)，a_0 为初始加速度(ft/s^2)，C 和 D 为常数，则车行驶的位移随时间的变化关系由下式确定：

$$x(t) = At^4 + Bt^3 + 2t^2$$

试用 lsqnonlin 函数估计出车速随时间变化的函数关系。已知车的初始位移、初始速度、初始加速度分别为 $x_0 = 0$ ft，$v_0 = 0$ ft/s，$a_0 = 2$ ft/s^2，计算最小二乘法产生的误差。如误差很大，试给出一种方法减小该误差。

[答案：$v(t) = -0.1084t^3 + 0.8307t^2 + 4t$。]

表 13 7　练习 13.9 中车辆的位移与行驶时间

位置(ft)	时间(s)
0	0
9	2 05
20	3 1
60	4 8
90	5 6
120	6 8

13.10　如图 13.19 所示，一个三杆桁架，以施力点的垂直偏移作为目标函数，[①] 如下所述：

$$\min f = \frac{Ph}{E}\frac{1}{x_1 + \sqrt{2}x_2}$$

其中杆的横截面积为 $A_1 = x_1$，$A_2 = x_2$，设计变量为 x_1，x_2，所施加力 P 的方向如图 13.19 中所示。

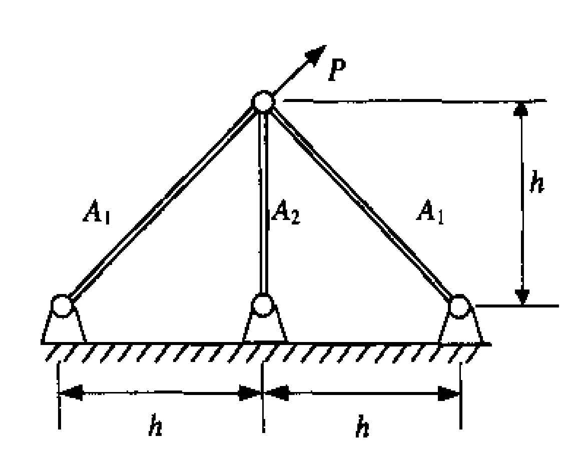


图 13.19　练习 13.10 中的三杆桁架

约束条件包括三杆实际应力的约束及设计变量的上下限约束，如下所示：

$$P\frac{x_2 + \sqrt{2}x_1}{\sqrt{2}x_1^2 + 2x_1 x_2} - \sigma^{(u)} \leqslant 0$$

① 参见 S S Rao，出处同上，第 530 页。

$$P\frac{1}{x_1+\sqrt{2}x_2}-\sigma^{(u)}\leqslant 0$$

$$P\frac{x_2}{\sqrt{2}x_1^2+2x_1x_2}+\sigma^{(l)}\leqslant 0$$

$$x_i^{(l)}\leqslant x_i\leqslant x_i^{(u)},\ i=1,2$$

式中 $\sigma^{(u)}$是张应力的最大值，$\sigma^{(l)}$为压缩应力的最大值。$x_i^{(l)}$ 为 x_i 的下限，$x_i^{(u)}$ 为 x_i 的上限。上述各参数值为 $\sigma^{(u)}=17.5,\sigma^{(l)}=-12,x_i^{(l)}=0.2,x_i^{(u)}=6.0(i=1,2),P=25,E=2,h=2$。用 fmincon 函数求出横截面及垂直偏移的最优解。已知初始条件$(x_1,x_2)=(0,0)$。

[答案：$[x_1,x_2]=[0.938,6.00]$和 $f=2.635$。]

13.11 如图 13.20 所示的电桥网络，有 5 个电阻 R_i，流过每个电阻的电流为 $I_i,i=1,\cdots,5$，电阻上的电压压降为 $V_i=R_iI_i$。假定每个电阻 R_i 的电压压降为一常数，即 $V_i=V_0$ 伏$(i=1,\cdots,5)$，且流过所有电阻的电流在下限 1 安培和上限 2 安培间变化。

(a)用公式描述一约束优化问题，从而求出总功率最小时的 R_i 值。

(b)用 fmincon 函数求出每个电阻的最优解及总功率值，已知 R_i 上的功率等于 $I_i^2R_i$，由基尔霍夫电流定律可知流入节点的所有支路电流的代数和为 0。

[答案：$[R_1,R_2,R_3,R_4,R_5]=[1,1,1.5,4,5]$，总功率 = 20 W。]

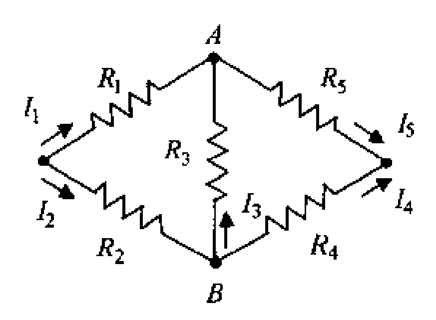


图 13.20 练习 13.11 的电桥网络

13.12 一公司有 m 个工厂用于生产某产品，该产品经轮船托运到 n 个仓库存放，第 j 处的仓库至少存 b_j 件产品，第 i 处的生产工厂能生产 a_i 件产品。从第 i 个工厂托运到仓库 j。x_{ij}件产品的运费为 $c_{ij}x_{ij}+d_{ij}x_{ij}^2$，式中 c_{ij} 和 d_{ij} 均为常数。上述问题可描述为二次规化问题，如下所示：

$$\min\sum_{i=1}^{m}\sum_{j=1}^{n}(c_{ij}x_{ij}+d_{ij}x_{ij}^2)$$

$$\text{s.t.}:\sum_{i=1}^{m}x_{ij}\geqslant b_j\qquad j=1,\cdots,n$$

$$\sum_{j=1}^{n}x_{ij}\leqslant a_i\qquad i=1,\cdots,m$$

$$x_{ij}\geqslant 0\qquad \text{对所有 } i,j \text{ 而言}$$

假定 $m=6, n=4, a=[8,24,20,24,16,12]', b=[29,41,13,21]'$，则：

$$c=\begin{bmatrix} 300\ 270\ 460\ 800 \\ 740\ 600\ 540\ 380 \\ 300\ 490\ 380\ 760 \\ 430\ 250\ 390\ 600 \\ 210\ 830\ 470\ 680 \\ 360\ 290\ 400\ 310 \end{bmatrix}$$

$$d=(-1)\begin{bmatrix} 7 & 4 & 6 & 8 \\ 12 & 9 & 14 & 7 \\ 13 & 12 & 8 & 4 \\ 7 & 9 & 16 & 8 \\ 4 & 10 & 21 & 13 \\ 17 & 9 & 8 & 4 \end{bmatrix}$$

试用 quadprog 函数求出由第 i 个工厂生产并托运到第 j 个仓库的产品数的最优解。[答案：$[x_{11}, x_{21}, x_{31}, x_{41}, x_{51}, x_{61}; x_{21}, x_{22}, x_{23}, x_{24}, x_{25}, x_{26}; x_{31}, x_{32}, x_{33}, x_{34}, x_{35}, x_{36}; x_{41}, x_{42}, x_{43}, x_{44}, x_{45}, x_{46}]' = [2.44,4.44,0,1.11,6.51,8.51;3.80,5.17,5.51,7.51,2.80,4.17;6.51,8.51,3.80,5.17,4.51,6.51;1.80,3.17,3.51,5.51,0.80,2.17]'$。]

13.13 如图 13.21 所示一个两杆桁架关于 y 轴对称，杆 1 和杆 2 的位置用无量纲值 x/h 表示，杆的横截面面积用无量纲值 A/A_{ref} 表示。此两项分别作为设计变量 x_1 和 x_2，其中 A_{ref} 为 A 的参考值，h 为桁架高度，所施力 P 的方向在 $-90°\leqslant\theta\leqslant 90°$ 间变化，要求计算出桁架重量 w 的最小值：

$$w=2phx_2A_{ref}\sqrt{1+x_1^2}$$

式中 ρ 为密度，约束条件包括杆 1 和杆 2 上的应力约束：

$$-\sigma_0\leqslant\frac{P\sqrt{1+x_1^2}(x_1\cos\theta-\sin\theta)}{2x_1x_2A_{ref}}\leqslant\sigma_0$$

$$-\sigma_0\leqslant\frac{P\sqrt{1+x_1^2}(x_1\cos\theta+\sin\theta)}{2x_1x_2A_{ref}}\leqslant\sigma_0$$

式中 P 为所施加外力，此外，设计变量 x_1, x_2 的下限和上限为：

$$x_i^{min}\leqslant x_i\leqslant x_i^{max}\qquad i=1,2$$

表 13.8 中列出各参数的相应值，试用 fseminf 函数求出设计变量的最优解并画出应力的曲线图。要求该应力为 θ 的函数，优化过程中采样间隔取 5°。若给定初始值：

$$(x_1^0, x_2^0)=(0.1,0.1)$$

结果如图 13.22 所示。

[答案 $[x_1, x_2]=[0.80,0.45], w=27.87$。]

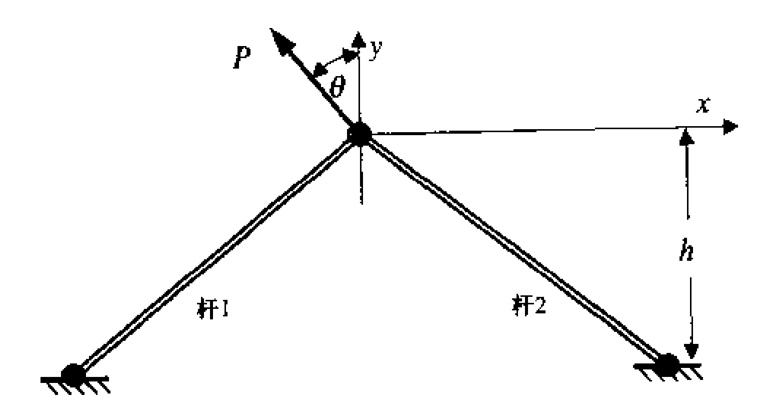


图 13 21 练习 13.13 的两杆桁架

表 13.8 练习 13.14 中的参数输入值

ρ(lb/in^3)	P(lb)	σ_0(psi)	h(in)	A_{ref}(in^2)	x_1^{min}	x_2^{min}	x_1^{max}	x_2^{max}
0 283	8 000	18 500	85	1	0 15	0 15	3 0	3 5

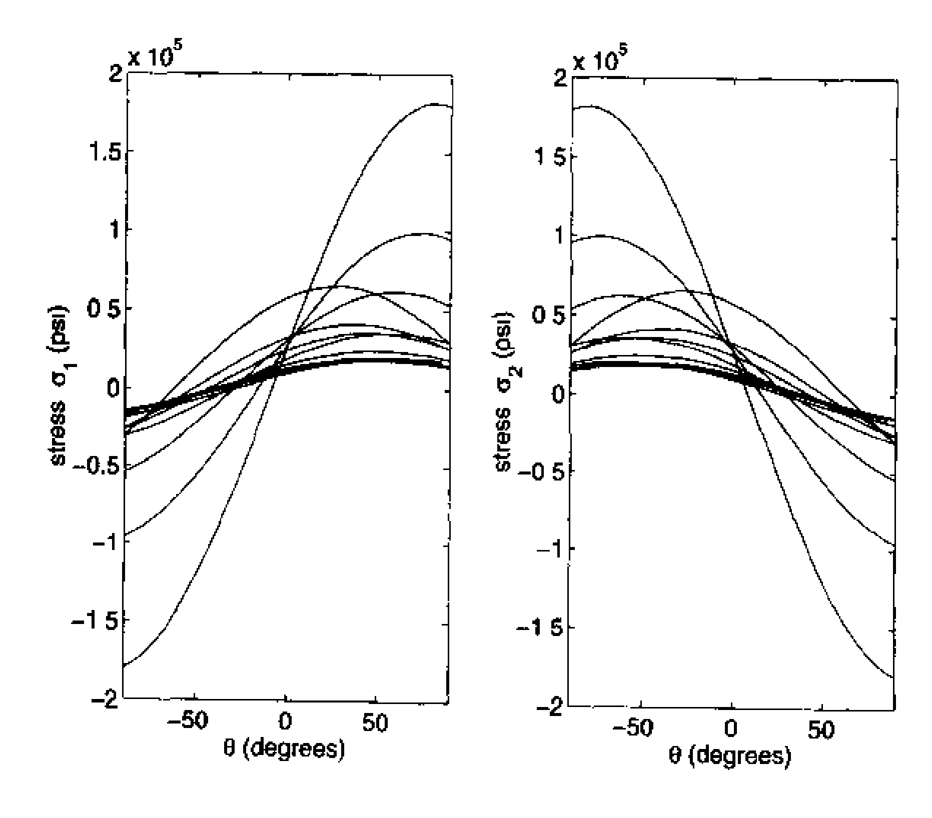


图 13.22 由优化过程得到的图 13.21 中的杆 1 和杆 2 的应力曲线图

参考文献

1. Arora J S. Introduction to Optimum Design. New York: McGraw-Hill, 1989.
2. Austin M and Chancogne D. Engineering Programming in C, MATLAB and JAVA. New York: John Wiley & Sons, 1998.
3. Branch M and Grace A. MATLAB Optimization Toolbox User's Guide. The Math Works. MA: Natick, 1996.

4. Changkong V and Haimes Y Y . Multiobjective Decision Making : Theory and Methodology . New York: Elsevier Science Publishing Co. ,1983.

5. Draper N and Smith H. Applied Regression Analysis. New York: John Wiley & Sons, 1966.

6. Eschenauer H, J Koski and Osyczka, Eds A. Multicriteria Design Optimization. New York: Springer-Verlag, 1990.

7. Golinski J. Optimum Synthesis Problems Solved by Means of Nonlinear Programming and Random
Methods. Mech. 5, 1970: pp. 287-309.

8. Gosselin C and Angeles J . A Global Performance Index for the Kinetic Optimization of Robotic Manipulators. ASMEJ Mech Design, 113, 1991: p. 222.

9. Rao S S. Engineering Optimization, Theory and Practice. 3rd ed. New York: John Wiley & Sons,
1996.

10. Kirsch U Optimal Structural Design. New York: McGraw-Hill, 1981.

11. Osyczka A. Multicriterion Optimization in Engineering with Fortran Programs. West Sussex: Ellis Horwood Ltd, 1984.

12. Papalambros P Y and Wilde D J. Principles of Optimal Design. Cambridge: Cambridge University Press, 1988.

13. Shigley J and Mischke C. Mechanical Engineering Design. New York: McGraw-Hill, 1989.

14. Stephenson D A and Agapiou J S. Metal Cutting Theory and Practice. New York: Marcel Dekker, 1997.

15. Vanderplaats G N Numerical Optimization Techniques for Engineering Design . New York : McGraw-Hill, 1984.

16. Wilde D J Globally Optimal Design. New York: John Wiley & Sons, 1978.

第14章 工程统计

本章介绍了 Statistics 工具箱在许多工程统计问题方面的应用。

14.1 统计量描述

有一组测量值 $x_j, j=1,2,\cdots,n$,其样本平均值为:

$$\bar{x} = \frac{1}{n}\sum_{j=1}^{n} x_j \tag{14.1}$$

样本方差为:

$$s^2 = \frac{1}{n-1}\left[\sum_{j=1}^{n} x_j^2 - n\bar{x}^2\right] \tag{14.2}$$

其中 s 为标准偏差。这里的 $\bar{x}$ 和 s 分别为平均值 μ 和标准偏差 σ 的估计值。求平均值的方法如下:

```
MeanValue = mean(x)
```

求标准偏差的方法如下:

```
StandardDeviation = std(x)
```

其中 $\boldsymbol{x}$ 为测量值向量或测量值矩阵。

将测量值分布的区域分为相等的 N 段,每段称为一区间。该区间内的 x_j 应大于等于其下限值,小于上限值,每个区间的中心元素用 $b_k, k=1,2,\cdots,N$ 表示。当所有 x_j 均分配到相应区间后,即可计算出该区间内的总数,用 n_k 表示,n_k 即为分布于中心元素为 b_k 的区间内数据的个数。如果把 n_k 看做其中心元素的一个函数,每段区间用一柱条表示,其宽度为区间上限与下限之差,这样得到的图称为直方图。每段区间内所含数据的个数由下述语句求得:

```
[nk,b] = hist(x,N)
```

其中 $\boldsymbol{nk}$ 为 n_k 的向量,$\boldsymbol{b}$ 为 hist 函数求出的区间中心值向量,x 为 n 个采样数据,N 为所分区间数。如果 N 值默认,则 MATLAB 将采用 $N=10$。不带返回值的 hist 函数用于画出直方图,其用法如下:

```
hist(x)
```

但 hist 函数常用于求 n_k。还可用:

```
bar(b,nk)
```

语句得到直方图。

定义 $f_k = n_k/n$,可得到分布于中心元素为 b_k 的区间内元素数与样本容量 n 的比值。令:

$$c_k = \sum_{j=1}^{k} f_j \qquad k = 1,2,\cdots,N$$

则称 c_k 为累积分布函数,可由下述函数求得:

```
ck = cumsum(f)
```

其中 $f = [f_1\ f_2 \cdots f_k] = [n_1/n\ \ n_2/n \cdots n_k/n]$。还可画出 c_k 对 b_k 的变化图,该图近似描述了一测量值小于或等于 x_k 的概率。

将 x_j 按从小到大顺序排序,最小值可用 min(x)求得,最大值由 max(x)求得,区间范围等于样本的最大值与最小值之差,可由语句:

```
rangex = range(x)
```

或:

```
rangex = max(x) - min(x)
```

求得。排序后的中心元素称为中位数。如果样本容量 n 为奇数,则中位数等于 $x_{(n+1)/2}$;如为偶数,则中位数等于 $(x_{n/2} + x_{n/2+1})/2$。中位数可由下述函数求得:

```
MedianValue = median(x)
```

另一种常用的统计量为几何均值。几何均值定义为样本中 n 个测量数据乘积的 n 次方根,即:

$$\bar{x}_g = \sqrt[n]{\prod_{j=1}^{n} x_j}$$

求几何均值的语句为:

```
GeometricMean = geomean(x)
```

或:

```
GeometricMean = prod(x)^(1/length(x))
```

对表 14.1 中的数据应用上述函数进行计算,得到结果如图 14.1 所示。表中数据放于函数 DataSet141 中,其定义为:

```
function d = DataSet141
d = [105 97 245 163 207 134 218 199.
    160 196 221 154 228 131 180 178...
    157 151 175 201 183 153 174 154...
    190 76 101 142 149 200 186 174...
    199 115 193 167 171 163 87 176...
    121 120 181 160 194 184 165 145...
    160 150 181 168 158 208 133 135...
    172 171 237 170 180 167 176 158...
    156 229 158 148 150 118 143 141...
    110 133 123 146 169 158 135 149];
```

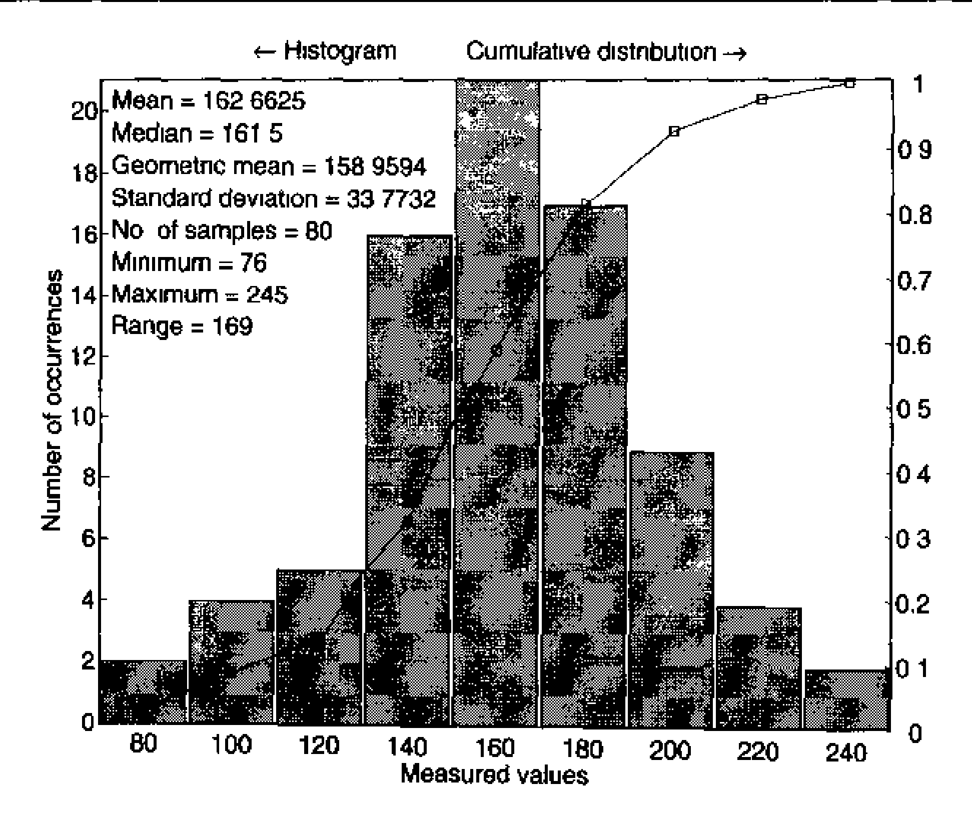


图 14.1 Dataset141 的数据直方图及累积分布函数曲线

表 14.1 由 DataSet141 构成的数据

105	97	245	163	207	134	218	199
160	196	221	154	228	131	180	178
157	151	175	201	183	153	174	154
190	76	101	142	149	200	186	174
199	115	193	167	171	163	87	176
121	120	181	160	194	184	165	145
160	150	181	168	158	208	133	135
172	171	237	170	180	167	176	158
156	229	158	148	150	118	143	141
110	133	123	146	169	158	135	149

程序清单为：

```
data = DataSet141;
ldat = length(data);
b = 80:20:240;
nn = hist(data,b);
maxn = max(nn);
cs = cumsum(nn * maxn/ldat);
bar(b,nn,.95,'y')
axis([70 250 0 maxn])
box off
hold on
plot(b,cs,'k-s')
title('\leftarrow Histogram      Cumulative distribution \rightarrow')
```

```
ylabel('Number of occurrences')
xlabel('Measured values')
text(72,.97 * maxn,['Mean = 'num2str(mean(data))])
text(72,.92 * maxn,['Median = 'num2str(median(data))])
text(72,.87 * maxn,['Geometric maen = 'num2str(geomean(data))])
text(72,.82 * maxn,['Standard deviation = 'num2str(std(data))])
text(72,.77 * maxn,['No.of samples = 'num2str(ldat)])
text(72,.67 * maxn,['Maximum = 'num2str(max(data))])
text(72,.72 * maxn,['Minimum = 'num2str(min(data))])
text(72,.62 * maxn,['Range = 'num2str(range(data))])
plot([70 250],[maxn maxn],'k',[250 250],[0 maxn],'k')
j=0:.1:1;
lenj = length(j);
text(repmat(251,lenj,1),maxn * j',num2str(j',2))
plot([repmat(248.5,1,lenj);repmat(250,1,lenj)],[maxn * j;maxn * j],'k')
```

尽管区间的中心值可由 hist 函数计算得出,但程序中仍然定义了这些值,以便更易于控制数据的输出形式。因为 box 函数分别在顶部和右部重复画出横轴和纵轴的刻度线,所以本程序中关闭了 box 函数。而且这里所画图形为两种不同类型的图,不能用 plotyy 函数实现,而必须使坐标标号独立于刻度线,从而分别画出顶部和右部图形边界线,由此得到的纵轴刻度最大值为 $21j$,即 y 轴的最大值为 21,可用 axis 函数求出。

如未定义区间中心值,则计算出的区间中心值将有所差别,由此画出的直方图也略有不同,该差别也改变了 n_k 值,因此,下述代码的执行结果如图 14.2 所示:

```
[nn,b] = hist(DataSet141,9);
bar(b,nn,.95,'y');
axis([70 250 0 max(nn)])
```

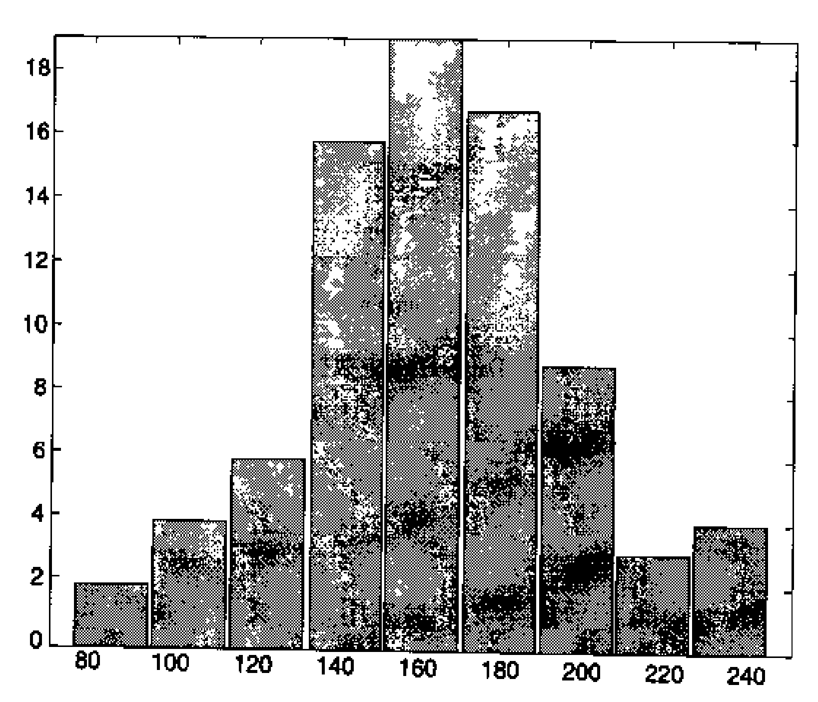


图 14.2　由 hist 函数计算区间中心值并分析 DataSet141 数据得到的直方图

图 14.1 和图 14.2 的差别是由区间中心的不同而引起的,第一种情况下区间的中心定义为:

```
b=[80 100 120 140 160 180 200 220 240]
```

而第二种情况由 hist 函数计算出的区间中心值为：

```
b=[85 38 104 16 122.94 141.72 160.50 179.27 198.05 216.83 235.61]
```

可以看出，在若干个区间内 x_j 的数量有所不同。

另一种方法是用盒状图表示数据。函数 DataSet141 的盒状图如图 14.3 所示，该图由下述语句画出：

```
boxplot(DataSet141,1)
```

函数的第二个参数设为 1，将产生一沙漏型盒状图，其凹口处为中位数。盒状图的顶部和底部间的区域表示了数据的 50%，其底部为第 1 分位数 q_1，顶部为第 3 分位数 q_3。注意一般盒状图并不是关于中位数对称的。分别从盒状图底部和顶部延伸出去的虚线代表极限值，其范围分别为 $q_1-1.5(q_3-q_1)$和 $q_3+1.5(q_3-q_1)$。位于虚线外的任何点被称为界外值，图中用加号表示。盒状图最常用来比较几组数据，如图 14.14(b)中所示。

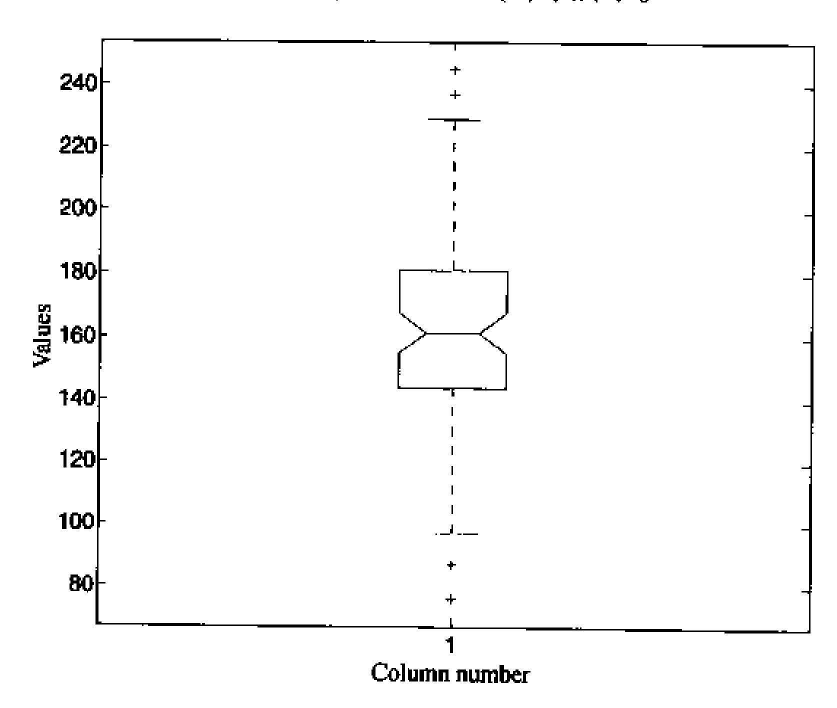


图 14.3 DataSet141 中数据的盒状图

q_1 和 q_3 的值分别由下列两式求得：

```
q1=prctile(DataSet141,25)
q3=prctile(DataSet141,75)
```

得出 $q_1=144$，$q_3=181$。prctile 函数的第 2 个参数用来定义所占比例。当比例等于 25% 时，即为第 1 分位数。

用下述语句可以确定数据分布平均值是否对称：

```
s=skewness (DataSet141)
```

返回值 $s=-0.0246$，负号意味着数据分布略倾向于左侧。

14.2 概率分布

14.2.1 离散分布

离散随机变量 $X=x$,其中 x 为变量 X 的取值,概率 $P(X)$定义为:

$$f(x) = P(X = x) \tag{14.3}$$

对所有 x,$f(x)\geqslant 0$ 且:

$$\sum_{\text{all } x_i} f(x_i) = 1 \tag{14.4}$$

$f(x)$被称为离散随机变量的概率质量函数。

如果只关注 $X\leqslant x$ 的概率 $P(X\leqslant x)$,则:

$$P(X \leqslant x) = \sum_{x_k \leqslant x} f(x_k) = 1 - \sum_{x_k > x} f(x_k) \tag{14.5}$$

称为累积分布函数。相反,如果只关注 $X\geqslant x$ 的概率 $P(X\geqslant x)$,则:

$$P(X \geqslant x) = \sum_{x_k \geqslant x} f(x_k) \tag{14.6}$$

二项分布 做 n 次重复试验,要求(1)每次试验是独立的,(2)每次试验只有两种可能:"成功"或"失败",(3)每次试验"成功"的概率 P 保持不变。该概率质量函数服从二项分布:

$$f_b(x) = P(X = x) = \frac{n!}{x!(n-x)!} p^x (1-p)^{n-x} \quad x = 0,1,\cdots,n \tag{14.7}$$

式中 x 为成功的试验次数。

二项分布的平均值为:

$$\bar{x} = np \tag{14.8a}$$

其标准偏差为:

$$s = \sqrt{np(1-p)} \tag{14.8b}$$

计算二项分布概率质量函数的 MATLAB 函数为:

```
binopdf(x,n,p)
```

计算平均值和方差(s^2)的函数为:

```
[Bmean, Bvariance] = binostat(n,p)
```

其中 $x=0,1,2,\cdots,n$。

在掷骰子的实验中,任一面成为顶面的概率为 $p=1/6$。考虑顶面点数为 3 的情况,投掷一次出现 3 点面的概率为:

$$P(X = \text{顶面为 3 点}) = \frac{1!}{1!(0!)}(1/6)^1(1-1/6)^{1-1} = \frac{1}{6}$$

可由下述语句实现:

```
h = binopdf(1,1,1/6)
```

而在两次投掷子中仅有一次顶面为 3 点面的概率为:

```
h=binopdf(1,2,1/6)
```

结果为 $h=0.2778<1/3$。

在投掷硬币的实验中，正反面的概率均为 $p=0.5$。投掷 10 次（$n=10$），出现 4 次为正面（$x=4$）的概率为：

```
h=binopdf(4,10,0.5)
```

结果为 $h=0.2051$。

例 14.1 飞机成功起飞的概率问题①

由 16 架飞机组成的空军飞行中队要求做好立即起飞的准备，其中一架飞机不能立即起飞的概率为 20%，重新起飞需几分钟的时间，因此一架飞机立刻起飞的概率为 0.80。12 架飞机能够成功起飞的概率为：

```
h=binopdf(12,16,.80)
```

计算结果为 $h=0.2001$。

另一方面，至少有 14 架飞机立刻成功起飞的概率[参考式(14.5)和式(14.6)]为：

```
h=1-binocdf(13,16,.80)   % 或 h=sum(binopdf(14:16,16,0.80))
```

计算结果为 $h=0.3518$。

下面为程序清单，其运行结果如图 14.4 所示。

```
n=1:16;
h=binopdf(n,16,.80);
plot([n;n],[zeros(1,16);h],'k')
text(8-.7:16-.7,h(8:16)+.005,num2str(h(8:16)',3))
axis([0 17 0 0.27])
xlabel('Number of aircraft launched on time')
ylabel('probability')
```

泊松分布 假定某一时间段内事件的发生是随机的，该时间段可分为多个更小的子时间段：(1)每个子时间段内事件发生多于一件的概率为零；(2)所有子时间段内事件发生的概率相同，且与子时间段长度成比例；(3)每个子时间段内发生的事件数独立于其他子时间段发生的事件数。这样的一系列事件发生过程叫做泊松过程。如果该时间段内发生事件总数的平均值 $\lambda>0$，则其概率质量分布为：

$$f_p(x)=\frac{e^{-\lambda}\lambda^x}{x!} \qquad x=0,1,2,\cdots \tag{14.9}$$

上式即为同一时间内发生 x 个事件的泊松分布。

泊松分布的平均值为：

$$\bar{x}=\lambda \tag{14.10a}$$

标准偏差为：

$$s=\sqrt{\lambda} \tag{14.10b}$$

① 参见 A J Hayter, *Probability and statistics for Engineers and Scientists*, PWS Publishing Co., Boston, 1996 年，第 167 页。

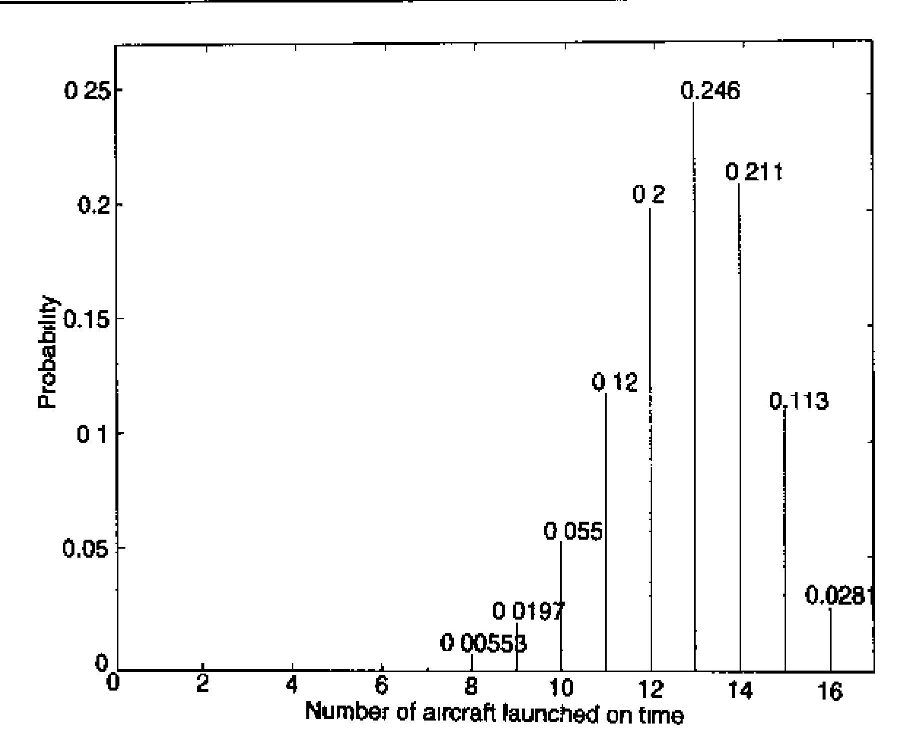


图 14.4　0～16 架飞机按时起飞的概率质量函数

泊松分布的概率质量函数由下述语句获得：

```
h = poisspdf(x,lambda)
```

计算平均值和方差(s^2)的语句为：

```
[Pmean, Pvariance] = poisstat(lambda)
```

例 14.2　医院资源储备

某医院的急救室平均每周收到 46 例心脏病发作病例(每天为 46/7 例)，目前医院每天能处理 9 例。医院的工作人员想知道其目前资源能否充足的概率，也就是 $P(X \leqslant 9)$ 的值。因此可用：

```
h = poisscdf(9,46/7)
```

求得结果为 $h = 0.8712$，由此看出每天还需增加 13% 的资源。

14.2.2　连续分布

假定 x_1, x_2 为连续随机变量 X 的上下限，则 X 位于 $x_1 \leqslant X \leqslant x_2$ 的概率 $P(X)$ 定义为：

$$P(x_1 \leqslant X \leqslant x_2) = \int_{x_1}^{x_2} f(u)\mathrm{d}u \tag{14.11a}$$

对所有 x 值，$f(x) \geqslant 0$，且：

$$\int_{-\infty}^{\infty} f(x)\mathrm{d}x = 1 \tag{14.11b}$$

$f(x)$ 被称为连续随机变量的概率密度函数(pdf)。

其累积分布函数(cdf) $F(x)$为:

$$F(x) = P(X \leqslant x) = \int_{-\infty}^{x} f(u)\mathrm{d}u = 1 - \int_{x}^{\infty} f(u)\mathrm{d}u \tag{14.12}$$

因此,

$$P(X \geqslant x) = 1 - F(x) \tag{14.13}$$

MATLAB 有一大批用于求概率密度的函数,现用其中的两个:正态分布和威布尔(Weibull)分布,其他函数用法与它们相似。

正态分布 正态概率分布函数为:

$$f_n(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad -\infty \leqslant x \leqslant \infty \tag{14.14}$$

式中 $-\infty < \mu < \infty$ 和 $\sigma > 0$ 为独立参数。μ 为正态分布的平均值,σ^2 为其方差(σ 为标准偏差)。如有一组数据 $x_j, j = 1,2,\cdots,n$,则其正态概率密度函数 pdf 由下述语句得到:

```
mu = mean(x);
sigma = std(x);
h = normpdf(x0,mu,sigma)
```

其中 $x0$ 为所关注的值(或者向量的多个值),$mu = \mu$, $sigma = \sigma$, h 的大小与 $x0$ 的大小相对应。μ 和 σ 的估计值可由下述语句得到:

```
[meanx, stddev] = normfit(x)
```

累积分布函数 $\Phi(x)$为:

$$\Phi(x) = P(X \leqslant x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} e^{-(u-\mu)^2/2\sigma^2}\mathrm{d}u = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{(x-\mu)/\sigma} e^{-u^2/2}\mathrm{d}u \qquad -\infty < x < \infty \tag{14.15}$$

或为:

$$\Phi(z) = P(Z \leqslant z) = \frac{1}{2\pi} \int_{-\infty}^{z} e^{-u^2/2}\mathrm{d}u \qquad -\infty < z < \infty \tag{14.16}$$

其中,

$$z = (x - \mu)/\sigma \tag{14.17}$$

称为标准正态随机变量,此时 $\mu_z = 0, \sigma_z = 1$。则:

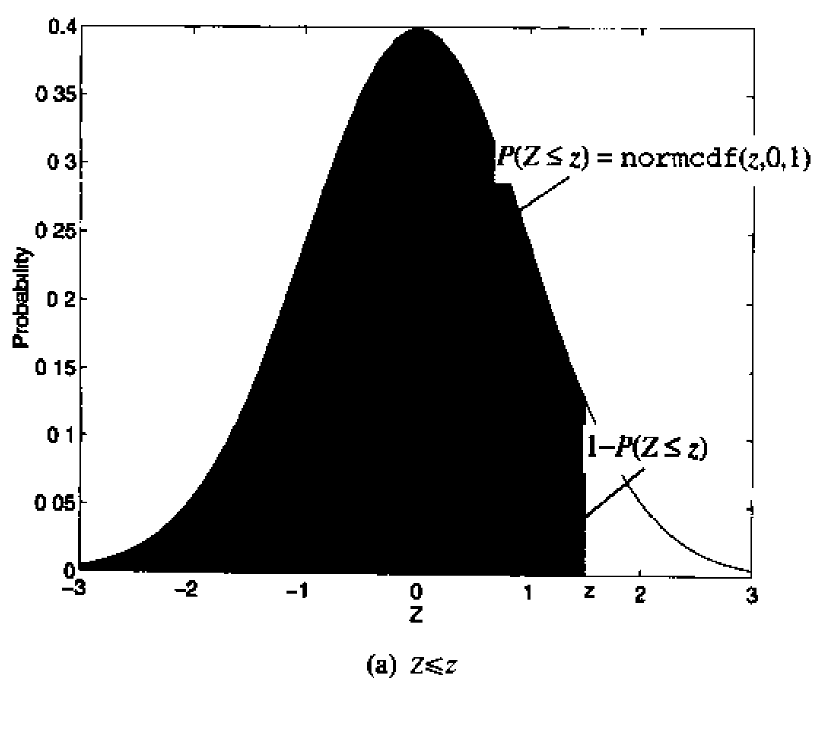
$$\begin{aligned} P(Z \geqslant z) &= 1 - \Phi(z) \\ P(Z \leqslant -z) &= \Phi(-z) \\ P(-z \leqslant Z \leqslant z) &= \Phi(z) - \Phi(-z) \\ P(z_L \leqslant Z \leqslant z_H) &= \Phi(z_H) - \Phi(z_L) \end{aligned} \tag{14.18}$$

式(14.18)中用 x 代替 z 也是正确的。由式(14.18)得到的区域如图 14.5 所示。

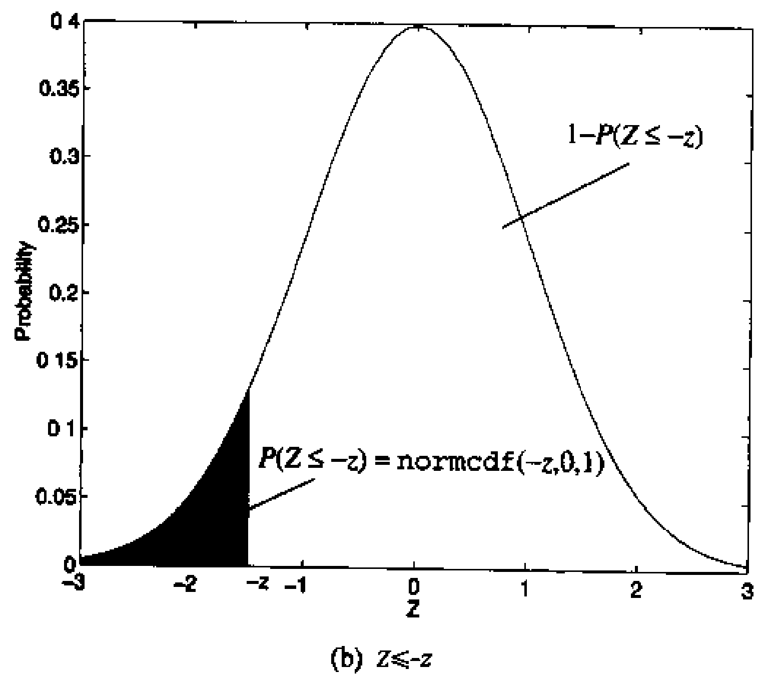
正态累积分布函数 cdf 由下列语句求得:

```
mu = mean(x);
sigma = std(x);
h = normcdf(x0,mu,sigma)
```

其中 $x0$ 为受关注的值(或向量),$mu=\mu$, $sigma=\sigma$, h 的大小与 $x0$ 的大小相对应。如果 x 转化为 z,则 $mu=0$, $sigma=1$,这些均为默认值。因此当 $x \to z$ 时,参数 mu 和 $sigma$ 可省略。



(a) $Z \leqslant z$



(b) $Z \leqslant -z$

图 14.5 由 normcdf 求出的 cdf 的关系图

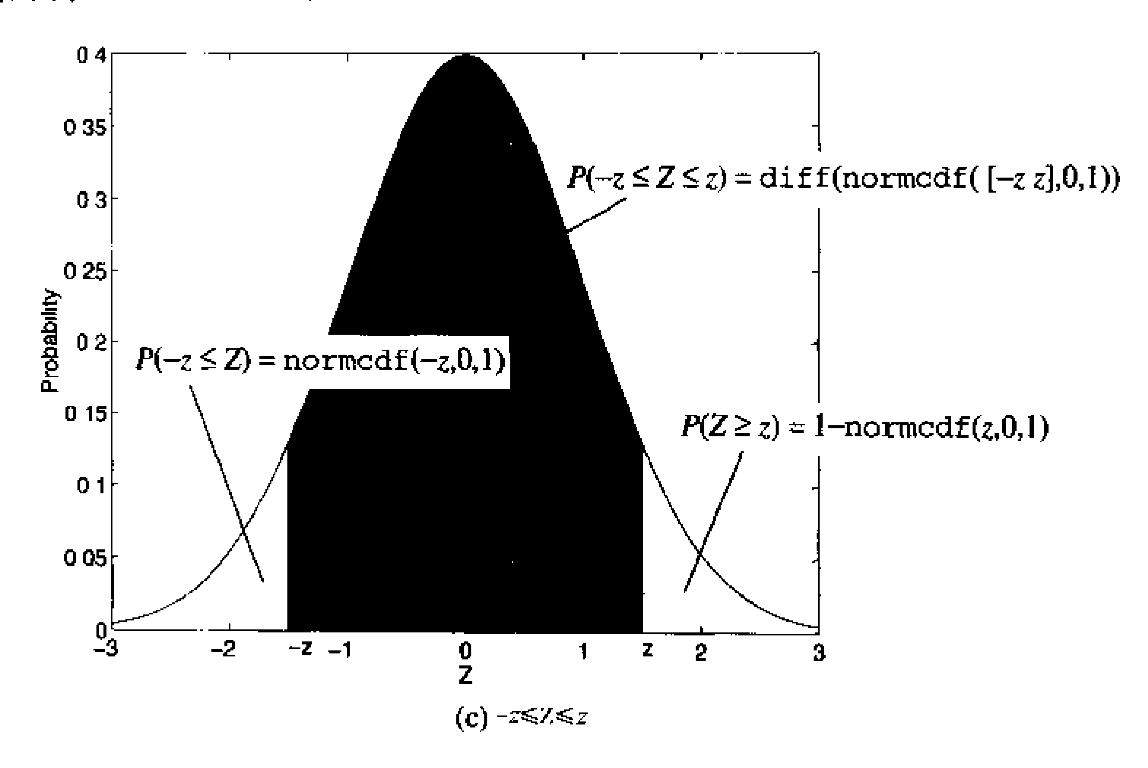
参考图 14.5(c)和式(14.18)可以得到 x 在一个取值范围内的概率分布 $P(x_L \leqslant X \leqslant x_H)$,即:

```
h=diff(normcdf([xL,xH],mean(x),std(x)))
```

例如,在函数 DataSet141 定义的数据中,介于 120 和 200 之间的测量数据的概率为:

```
h=diff(normcdf([120 200],mean(DataSet141),std(DataSet141)))
```

其结果为 $h=0.7623$。此值与图 14.1 中得到的估计值非常接近。



(c) $-z \leqslant Z \leqslant z$

图 14.5(续图) 由 normcdf 求出的 cdf 的关系图

在一些例子中，有时希望求出 $\Phi(x)$的反函数，即：

$$x = \Phi^{-1}[p(X \leqslant x)] \tag{14.19}$$

可由下述函数实现：

```
x = norminv(p, mean(x),std(x))
```

其中 p 为累积概率，如图 14.5(a)中的阴影部分所示。

为确定一组数据能否用正态概率密度函数表示，通常需在正态概率图中画出这些数据。正态概率图的纵坐标(y 轴)的刻度由累积正态分布函数划分。在正态概率坐标上画图与在对数刻度坐标上画图类似，在对数坐标图上指数函数将显示为一条直线；与之类似，在以正态累积概率函数坐标上画出服从正态分布的过程将显示为一条直线。换句话说，位于平均值 μ 两侧标准偏差 σ 处的累积概率值服从正态分布，其值分别为 $P(X \leqslant \mu+\sigma)=0.84$ 和 $P(X \leqslant \mu-\sigma)=0.16$。平均值处的累积概率值 $P(x \leqslant \mu)=0.5$，转换到概率图上的三组坐标($\mu-\sigma$,0.16)，(μ,0.5)和($\mu+\sigma$,0.84)对应的三个点位于一条直线上。μ 和 σ 分别由式(14.1)和式(14.2)估算，并用 MATLAB 函数 mean 和 std 计算。

在概率坐标上画图的过程如下所述。有 m 个数据 y_i，$i=1,2,\cdots,m$，将它们按照从小(最负值)到大(最大正数)顺序排序，给最小值编号为 1，依次为 2，3 等等，最大值编号为 m。这些排好序的数据表示为 w_j，$j=1,2,\cdots,m$。对每个指定累积概率 w_j，$j=1,2,\cdots,m$，即概率 $P(w \leqslant w_j)$。概率图所画数据值的坐标为$(j-0.5)/m$。如果只有线性坐标图，该点坐标变为(w_j，z_j)，其中 $z_j=\text{norminv}((j-0.5)/m)$。执行上述计算并画出图形的函数为：

```
normplot(y)
```

其中 $y=[y_1\ y_2 \ldots\ y_m]$。图中直线由 y_i 和 z_j 的第1分位数和第3分位数组成的坐标确定,可参照第14.1节中的 $q1$ 和 $q3$ 的定义及图14.3中的解释。

例 14.3 正态分布数据的验证

再次利用 DataSet141 函数所定义的数据重新画出对应的直方图,并在柱条上重叠画出对应的正态概率密度函数曲线。可由下述语句实现:

```
histfit(DataSet141,9)
colormap([1 1 1])
```

结果如图14.6所示。然后验证数据是否服从正态分布。用下述语句可得到如图14.7所示的结果:

```
normplot(DataSet141)
whitebg('white')
```

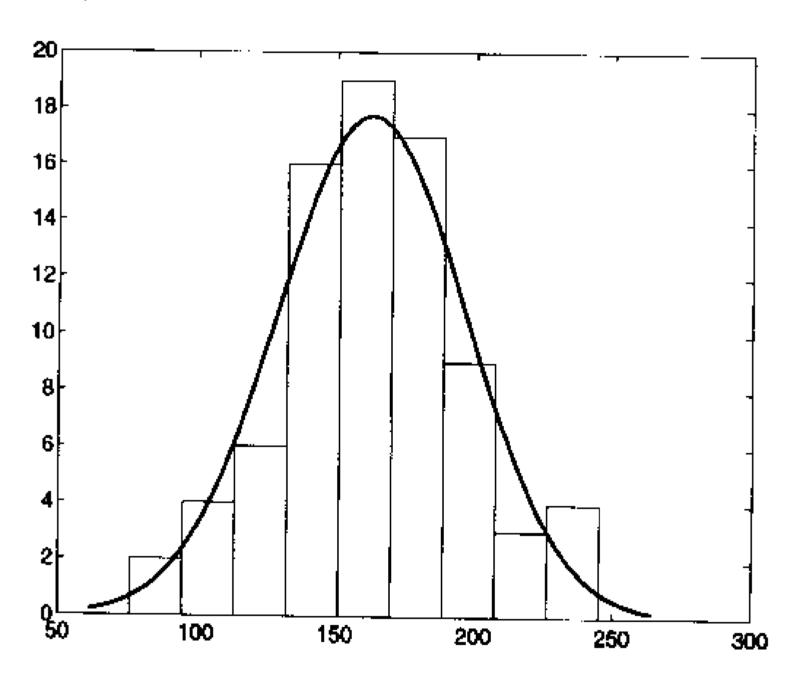


图 14.6 叠加正态概率密度函数(pdf)曲线直方图

从图中可看出有相当多的数据位于直线附近,可以得出结论认为该数据近似服从正态分布。如果承认这些数据服从正态分布,则90%的数据分布值可确定出来,参照图14.5(a)和图14.5(c)以及式(14.19),用 norminv 函数求解如下:

```
zh = norminv(.95, mean(DataSet141),std(DataSet141))
zl = norminv(.05, mean(DataSet141),std(DatSet141))
```

得到结果为 $z_h=218.2145$ 及 $z_l=107.1105$。为验证此结果,现求出这两个界限概率之差,如下所述:

```
ph = normcdf (218.2145, mean(DataSet141),std(DataSet141))
pl = normcdf (107.1105, mean(DataSet141),std(DataSet141))
```

执行结果为:$p_h-p_l=0.9500-0.0500=0.90$

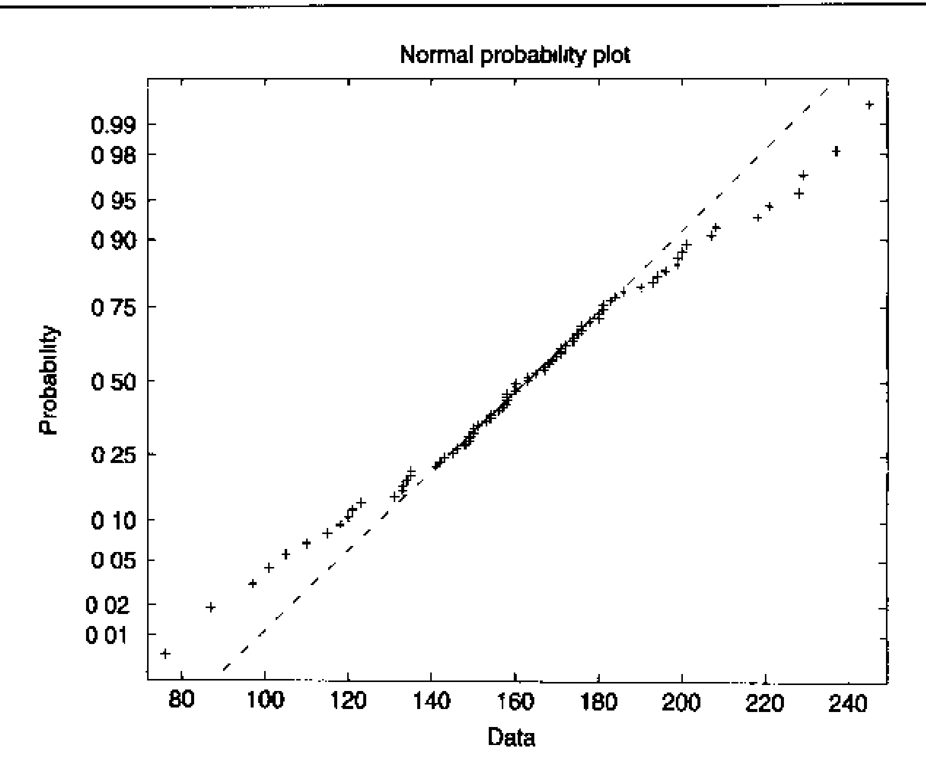


图 14.7　DataSet141 所定义数据的正态累积概率图

当 $np>5$ 且 $n(1-p)>5$ 时，正态分布非常近似于二项分布，当 $\lambda>5$ 时 $P(X\leqslant x)$ 的分布也非常近似于泊松分布。对于二项分布，标准正态随机变量为[参考式(14.8)]：

$$Z_b = \frac{X - np}{\sqrt{np(1-p)}}$$

对于泊松分布[参考式(14.10)]为：

$$Z_p = \frac{X - \lambda}{\sqrt{\lambda}}$$

为说明泊松分布与正态分布的近似性，再参见例 14.2。例 14.2 中得出某医院每天治疗 9 例以下心脏病患者的概率为 0.871 2。现用正态分布重新计算，已知 $\mu = 46/7$ 及 $\sigma = \sqrt{46/7}$，则：

```
h = normcdf((9 - 46/7)/sqrt(46/7),0,1)
```

结果为 $h=0.828\,3$，在精确值的 4.9% 范围内。为直观地显示出这种近似关系，运行下面的程序，它可以绘出例 14.2 中的泊松分布图形以及由上述表达式得到的近似图形：

```
x = 1:16;
y = 0:.2:16;
yPoisson = poisspdf(x,46/7);
NormApprox = normpdf(y,46/7,sqrt(46/7));
plot([x,x],[zeros(1,16);yPoisson],'k',y,NormApprox,'k')
xlabel('Number of aircraft launched on time')
ylabel('Probability density')
```

其执行结果如图 14.8 所示。

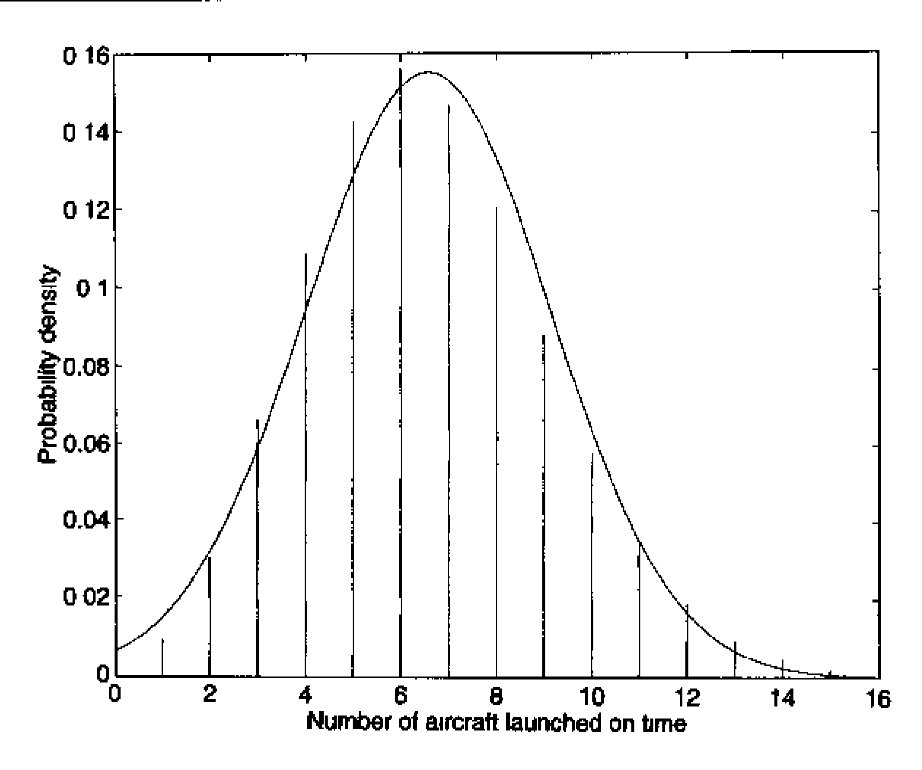


图 14.8 例 14.1 中的泊松分布及其近似的正态分布图

威布尔分布 威布尔概率分布函数为：

$$f_w(x) = \alpha\beta x^{\beta-1} e^{-\alpha x^\beta} \quad x > 0 \tag{14.20}$$

式中 α>0 为尺度参数，β>0 为形状参数（另一种常用表示形式为经 $\alpha = \delta^{-\beta}$ 变换后得到的形式）。当 β=1 时，式(14.20)变为指数分布；β=2 时，变为瑞利(RayLeigh)分布。该概率密度函数的平均值为：

$$\mu = \alpha^{-1/\beta}\Gamma\left(1 + \frac{1}{\beta}\right)$$

方差为：

$$\sigma^2 = \alpha^{-2/\beta}\Gamma\left(1 + \frac{2}{\beta}\right) - \alpha^{-2/\beta}\left[\Gamma\left(1 + \frac{1}{\beta}\right)\right]^2$$

式中 $\Gamma(x)$ 为伽马函数。威布尔概率密度函数由下述语句实现：

```
h = weibpdf(x, alpha, beta)
```

其中，alpha = α，beta = β，h 的大小与 x 的大小相对应。其平均值和方差由下述语句得到：

```
[Wmean, Wvariance] = weibstat(alpha, beta)
```

累积分布函数 $F_w(x)$ 为：

$$F_w(x) = P(X \leq x) = 1 - e^{-\alpha x^\beta} \quad x > 0 \tag{14.21}$$

则由下述语句得到：

```
h = weibcdf(x, alpha, beta)
```

其中 h 的大小与 x 的大小相对应。

有时需求出 $F(X)$ 的反函数，即：

$$x = F_w^{-1}[P(X \leqslant x)] \tag{14.22}$$

可用下述函数实现：

```
x = weibinv(p,alpha,beta)
```

其中 p 为累积概率分布值。

例 14.4　对服从威布尔分布的数据进行验证

表 14.2 所列为元件寿命的分类数据，下面创建函数 DataSet142 来存放这些数据：

```
function d = DataSet142
d = [72 82 97 103 113 117 126 127 127 139 154 159 199 207]';
```

在图上画出这些数据以确定是否能用威布尔分布描述。用 weibplot 函数实现，其结果如图 14.9 所示：

```
weibplot (DataSet142)
whitebg ('white')
```

由图中可看出，这些数据用威布尔分布描述非常合适，因此将采用威布尔分布模型来表示这些数据。

表 14.2　元件寿命分类数据

元件寿命	$F(t)=(j-0.5)/14$	j
72	0.0357	1
82	0.1071	2
97	0.1786	3
103	0.2500	4
113	0.3214	5
117	0.3929	6
126	0.4643	7
127	0.5357	8
127	0.6071	9
139	0.6786	10
154	0.7500	11
159	0.8214	12
199	0.8929	13
207	0.9643	14

首先确定 α 和 β 值。先对式(14.21)加以变换，然后取两次自然对数得到下式：

$$y = \beta x + b \tag{14.23}$$

式中，

$$y = \ln\{\ln(1/[1 - F(t)])\}$$
$$x = \ln(t) \qquad b = \ln(\alpha) \tag{14.24}$$

因此，

$$\alpha = e^b \tag{14.25}$$

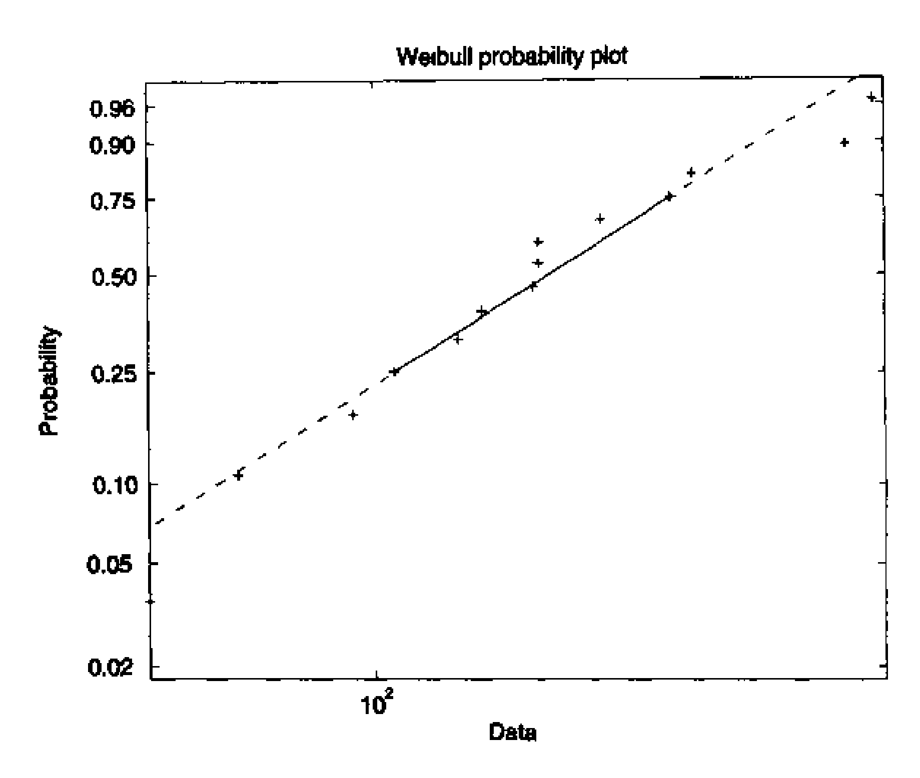


图 14.9 DataSet142 数据的威布尔累积概率图

式(14.23)表示一直线方程，其斜率 β 和截距 b 可由函数 polyfit 计算。β 和 b 一旦确定，α 可由式(14.25)确定，$F(t)$ 函数值用例 14.3 中的方法解出，其值列于表 14.2 第 2 列上。程序清单为：

```
x = log(DataSet142)'
y = log(log(1./(1 - ((1:14) - .5)/14)));
c = polyfit(x,y,1);
[Wmean, Wvariance] = weibstat(exp(c(2)),c(1));
disp(['beta = 'num2str(c(1))' alpha = 'num2str(exp(c(2)))])
disp(['Mean = 'num2str(Wmean)' Std dev = 'num2str(sqrt(Wvariance))])
```

执行程序后，在 MATLAB 命令窗口上将显示下列数据：

```
beta = 3.9862        alpha = 2.5234e-009
Mean = 130.0703      Std dev = 36.6047
```

假定为一正态分布，则：

```
Nmean = mean(DataSet142)
Stddev = std(DataSet142)
```

运行结果为 $\mu_{norm} = 130.1429$ 和 $\sigma_{norm} = 39.3854$。

运行下面的代码将画出威布尔概率密度函数和正态密度函数的关系图，以便于两者比较。程序清单如下：

```
x = log(DataSet142)';
y = log(log(1./(1 - ((1:14) - .5)/14)));
c = polyfit(x,y,1);
xx = 50:2.5:200;
```

```
yW=weibpdf(xx,exp(c(2)),c(1));
yN=normpdf(xx,mean(DataSet142),std(DataSet142));
plot(xx,yW,'k-',xx,yN,'k--')legend('Weibull','normal')
xlabel('x')
ylabel('Probability density function')
```

执行结果如图 14.10 所示。

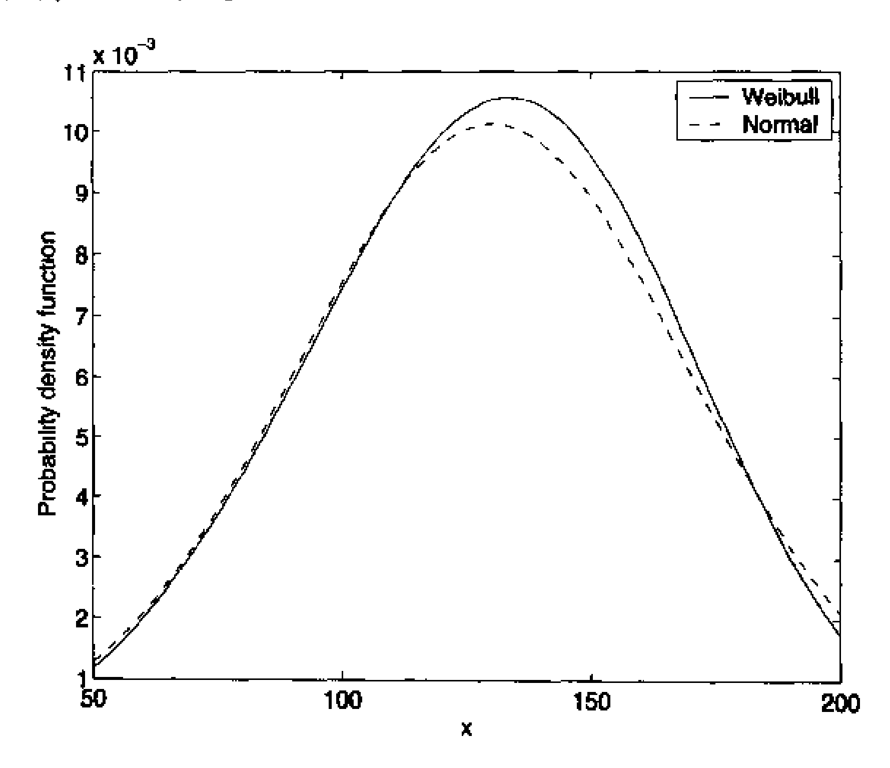


图 14.10 由 DataSet142 数据得出的威布尔函数和正态概率密度函数的比较

现在要求确定元件寿命少于 100 小时的概率，程序清单如下：

```
x=log(DataSet142)';
y=log(log(1./1-((1:14)-.5)/14)));
c=polyfit(x,y,1);
p=weibcdf(100,exp(c(2)),c(1))
```

其结果为 $p=0.2109$ 或 21%。

服从指数分布和对数分布的数据也需要用到变换技术，即将累积分布函数变换为直线方程。对于这种情况 MATLAB 没有明确给出 normplot 和 weibplot 的函数，但是这两种情况均可作为使用 weibplot 函数和 normplot 函数的特例(有关对数分布的情况可参考练习 14.11)。

14.3 置信区间

令 θ 为 n 个数据样本中的某一统计量(例如平均值、方差、均差等)，人们所关心的是下式中的 l 和 u 的值：

$$p(l \leqslant \theta \leqslant u) = 1 - \alpha$$

式中 $0<\alpha<1$。上式说明，在容量为 n 的样本中找出包含 θ 真值的区间的概率为 $1-\alpha$。该区间：

$$l \leqslant \theta \leqslant u$$

称做置信度为 100(1 - α)% 的θ的双限置信区间。l 和 u 分别称做置信上限和置信下限。与之类似，置信度为 100(1 - α)% 的下限置信区间为：

$$l \leqslant \theta$$

置信度为 100(1 - α)% 的上限置信区间为：

$$\theta \leqslant u$$

置信上限和下限值取决于样本的分布，还与总体标准偏差是否为已知有关。表 14.3 中总结了几种常用的确定置信上下限的关系式，其中用到下述定义：

μ 和 σ 分别为真实平均值和标准偏差值
$\bar{x}$ 和 s^2 分别由式(14.1)和式(14.2)确定
$t_{\alpha,n-1}$为由函数 tinv 求得的服从 $n-1$ 自由度的 t 分布值
$z_{\alpha/2}$为由函数 norminv 求得的正态分布值
$\chi^2_{\alpha/2,n-1}$为由函数 chi2inv 得到的 $n-1$ 自由度的 X^2 分布值
$f_{\alpha/2,n-1,m-1}$为由函数 finv 得到的 $n-1$ 自由度和 $m-1$ 自由度的 f 分布值

第 4 种情况中的 s_p 为：

$$s_p = \sqrt{\frac{(n_1-1)s_1^2 + (n_2-1)s_2^2}{n_1+n_2-2}}$$

第 5 种情况中的 v 为：

$$v = \left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2 \left[\frac{(s_1^2/n_1)^2}{n_1+1} + \frac{(s_2^2/n_2)^2}{n_2+1}\right]^{-1} - 2$$

表 14.3　几种置信区间过程的总结

问题类型	统计量	100(1 - α)% 置信区间 $\hat{\theta} - q \leqslant \theta \leqslant \hat{\theta} + q$			情况
	$\hat{\theta}$	θ	q		
已知 σ^2 的平均值问题	$\bar{x}$	μ	$z_{\alpha/2}\sigma/\sqrt{n}$		1
已知 σ_1^2 和 σ_2^2 的均差问题	$\bar{x}_1 - \bar{x}_2$	$\mu_1 - \mu_2$	$z_{\alpha/2}\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}$		2
未知 σ^2 的平均值问题	$\bar{x}$	μ	$t_{\alpha/2,n-1}s/\sqrt{n}$		3
未知 $\sigma_1^2 = \sigma_2^2$ 的均差问题	$\bar{x}_1 - \bar{x}_2$	$\mu_1 - \mu_2$	$t_{\alpha/2,n_1+n_2-2}s_p\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}$		4
未知 $\sigma_1^2 \neq \sigma_2^2$ 的均差问题	$\bar{x}_1 - \bar{x}_2$	$\mu_1 - \mu_2$	$t_{\alpha/2,v}\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}$		5
		100(1 - α)% 置信区间 $q_1\hat{\theta} \leqslant \theta \leqslant q_2\hat{\theta}$			
	$\hat{\theta}$	θ	q_1	q_2	
方差	S^2	σ^2	$\frac{n-1}{\chi^2_{\alpha/2,n-1}}$	$\frac{n-1}{\chi^2_{1-\alpha/2\ n-1}}$	6
方差比值	$\frac{s_1^2}{s_2^2}$	$\frac{\sigma_1^2}{\sigma_2^2}$	$\frac{1}{f_{\alpha/2,n_1-1,n_2-1}}$	$f_{\alpha/2,n_2-1,n_1-1}$	7

现以表 14.3 中第 3 种情况和第 7 种情况为例说明确定置信上限和下限的方法。对第 3 种情况,双限置信区间为:

$$\bar{x} - t_{a/2,n-1} s/\sqrt{n} \leqslant \mu \leqslant \bar{x} + t_{a/2,n-1} s/\sqrt{n}$$

对于第 7 种情况,双限置信区间为:

$$\frac{s_1^2}{s_2^2}\frac{1}{f_{a/2,n_1-1,n_2-1}} \leqslant \frac{\sigma_1^2}{\sigma_2^2} \leqslant \frac{s_1^2}{s_2^2} f_{a/2,n_2-1,n_1-1}$$

注意上式中 f 的自由度顺序是相反的。

第 3 种情况:

对于第 3 种情况,考虑表 14.1 中的数据,该数据放于函数 DataSet141 内。如令置信度为 95%,求置信区间的程序如下所述:

```
meen = mean(DataSet141);
L = length(DataSet141);
q = std(DataSet141) * tinv(0 975,L - 1)/sqrt(L);
disp(['Sample mean = 'num2str(meen)])
disp('Confidence interval for sample mean at 95% confidence level -')
disp([' 'num2str(meen-q)' < = Sample mean < = 'num2str(meen + q)])
```

执行结果在 MATLAB 命令窗口中显示下列数据:

```
Sample mean = 162.6625
Confidence interval for sample mean at 95% confidence level-
  155.1466 < = Sample mean < = 170.1784
```

另一个求置信区间的函数为 ttest,将在第 14.4 节介绍。

第 7 种情况:

对于第 7 种情况,考虑表 14.4 中的两列数据,这些数据由函数 DataFci 定义如下:

```
function [set1,set2] = DataFci
set1 = [41.60 41.48 42.34 41.95 41.86 42.18 41.72 42.26 41.81 42.04];
set2 = [39.72 42.59 41.88 42.00 40.22 41.07 41.90 44.29];
```

表 14.4 第 7 种情况中用到的数据

Set 1	Set 2
41.60	39.72
41.48	42.59
42.34	41.88
41.95	42.00
41.86	40.22
42 81	41 07
41.72	41.90
42.26	44.29
41.81	
42 04	

程序清单如下：

```
[data1,data2] = DataFcn;
r = var(data1)/var(data2);
L1 = length(data1);
L2 = length(data2);
q2 = r * finv(.975,L2-1,L1-1);
q1 = r/finv(.975,L1-1,L2-1);
disp(['Ratio of sample variances = 'num2str(r)])
disp('Confidence interval for ratio of sample variances at 95% confidence level -')
disp([' 'num2str(q1)' <= Ratio of sample variances <= 'num2str(q2)])
```

执行后在 MATLAB 命令窗口中显示如下数据：

```
Ratio of sample variances = 0.039874
Confidence interval for ratio of sample variances at 95% confidence level -
  0.0082672 <= Ratio of sample variances <= 0.16736
```

14.4 假设检验

在工程上，许多情况下需要接受或拒绝某些参数的描述（假设）。统计假设可以认为是对一个或多个总体参数的描述，总体是指所重点观察的全部，而单个样本是总体的子集。如果用概率分布表示总体，则统计假设可被认为是对总体的统计分布的描述。

假定参数 θ 是从某个总体的 n 个样本中得到的统计量，现在要判断 θ 是否等于 θ_0。假设检验过程可分为如下步骤：

1. 提出原假设 H_0
2. 建立合适的检验统计量 q_0
3. 选择置信度[θ 的置信度为 $100(1-\alpha)\%$]
4. 把检验统计量 q_0 与期望自然发生的统计量 q 的值相比较

根据 q_0 和 q 的不同数量级可以确定接受或是放弃原假设。如果放弃原假设，则接受了另一假设，用 H_1 表示。

有下列三种情况需要考虑：

$H_0:\theta=\theta_0$　　$H_0:\theta=\theta_0$　　$H_0:\theta=\theta_0$

$H_1:\theta\neq\theta_0$　　$H_1:\theta>\theta_0$　　$H_1:\theta<\theta_0$

每种情况对应统计量 $q_0(n,\alpha)$ 和 $q(n,\alpha)$。表 14.5 总结了几种对假设的检验过程，与表 14.3 中所列的置信区间过程相对应。表 14.5 中各个量的定义可参考第 14.3 节。

在假设检验中可能会得出两种类型的错误结论：

第 1 种错误：原假设 H_0 为真时被放弃

第 2 种错误：原假设 H_0 为假时被接受——此时 $\theta=\theta_1$

产生第 1 种错误的概率用 α 表示，产生第 2 种错误的概率用 β 表示。

现在普遍用一种称做 p 值的量代替置信参数 α。p 值越小，原假设成立的可能性越小。现用表 14.5 的情况 2、情况 4 和情况 7 来解释这些概念。

表 14.5 几种假设检验过程

原假设 H_0	另一种假设 H_1	放弃 H_0 的条件	检验设计	MATLAB 函数	情况
$\mu=\mu_0$ (已知 σ)	$\mu\neq\mu_0$	$\lvert z_0\rvert > z_{\alpha/2}$	$z_0=\frac{\bar{x}-\mu_0}{\sigma/\sqrt{n}}$	ztest	1
	$\mu>\mu_0$	$z_0>z_\alpha$			
	$\mu<\mu_0$	$z_0<-z_\alpha$			
$\mu=\mu_0$ (未知 σ)	$\mu\neq\mu_0$	$\lvert t_0\rvert > t_{\alpha/2,n-1}$	$t_0=\frac{\bar{x}-\mu_0}{s/\sqrt{n}}$	ttest	2
	$\mu>\mu_0$	$t_0>t_{\alpha,n-1}$			
	$\mu<\mu_0$	$t_0<-t_{\alpha,n-1}$			
$\mu_1=\mu_2$ (已知 σ_1 和 σ_2)	$\mu_1\neq\mu_2$	$\lvert z_0\rvert > z_{\alpha/2}$	$z_0=\frac{\bar{x}_1-\bar{x}_2}{\sqrt{\frac{\sigma_1^2}{n_1}+\frac{\sigma_2^2}{n_2}}}$		3
	$\mu_1>\mu_2$	$z_0>z_\alpha$			
	$\mu_1<\mu_2$	$z_0<-z_\alpha$			
$\mu_1=\mu_2$ (未知 $\sigma_1=\sigma_2$)	$\mu_1\neq\mu_2$	$\lvert t_0\rvert > t_{\alpha/2,n_1+n_2-2}$	$t_0=\frac{\bar{x}_1-\bar{x}_2}{s_p\sqrt{\frac{1}{n_1}+\frac{1}{n_2}}}$	ttest2	4
	$\mu_1>\mu_2$	$t_0>t_{\alpha,n_1+n_2-2}$			
	$\mu_1<\mu_2$	$t_0<t_{\alpha,n_1+n_2-2}$			
$\mu_1=\mu_2$ (未知 $\sigma_1\neq\sigma_2$)	$\mu_1\neq\mu_2$	$\lvert t_0\rvert > t_{\alpha/2,v}$	$t_0=\frac{\bar{x}_1-\bar{x}_2}{\sqrt{\frac{s_1^2}{n_1}+\frac{s_2^2}{n_2}}}$		5
	$\mu_1>\mu_2$	$t_0>t_{\alpha,v}$			
	$\mu_1<\mu_2$	$t_0<-t_{\alpha,v}$			
$\sigma^2=\sigma_0^2$	$\sigma^2\neq\sigma_0^2$	$\chi_0^2>\chi^2_{\alpha/2,n-1}$	$\chi_0^2=\frac{(n-1)s^2}{\sigma_0^2}$		6
	$\sigma^2>\sigma_0^2$	$\chi_0^2>\chi^2_{\alpha,n-1}$			
	$\sigma^2<\sigma_0^2$	$\chi_0^2<\chi^2_{1-\alpha,n-1}$			
$\sigma_1^2=\sigma_2^2$	$\sigma_1^2\neq\sigma_2^2$	$f_0>f_{\alpha/2,n_1-1,n_2-1}$ 或 $f_0<f_{1-\alpha/2,n_1-1,n_2-1}$	$f_0=\frac{s_1^2}{s_2^2}$		7
	$\sigma_1^2>\sigma_2^2$	$f_0>f_{\alpha,n_1-1,n_2-1}$			

情况 2：

用表 14.1 中函数 DataSet141 包含的数据对情况 2 加以分析。在置信度为 95% 时确定样本平均值与数值 168($\mu_0=168$)之间是否存在显著性差异。这里假设：

$$H_0:\mu=168$$

$$H_1:\mu\neq168$$

用函数 ttest 分析该假设的合理性。函数 ttest 的用法如下：

```
[h,p,ci] = ttest(Data,muzero,alpha)
```

其中 *Data* 为数据，*muzero* $=\mu_0$，*alpha* $=\alpha$，H_0 成立时 $h=0$，H_1 成立时 $h=1$。对双限置信区间 p 值的计算方法为：

```
p=2*(1-tcdf(t0,n-1));
```

其中 $t0=t_0$ 在表 14.5 中第 2 行第 4 列定义，$ci(1)=l$ 及 $ci(2)=u$ 分别为置信下限和置信上限。则：

```
[h,p,ci] = ttest(DataSet141,168,0 05)
```

其结果为 $h=0$,即不能放弃原假设,$p=0.161\,4$, $ci(1)=155.146\,6$, $ci(2)=170.178\,4$。在第14.3节的情况 3 中,已求出 $\bar{x}=162.662\,5$,而且当置信度为95%时,置信区间为 $155.146\,6\leqslant\bar{x}\leqslant170.178\,4$。因为假设平均值取 168,该数值位于上述置信区间内,所以应该推断出原假设将不应被放弃。事实上,由 p 值也可以看出只有 $100(1-0.161\,4)=83.9\%$ 的置信度,比期望值 95% 低。另一方面,如原假设为:

$$H_0:\ \mu=175$$

$$H_1:\ \mu\neq175$$

则:

```
[h,p,ci] = ttest(DataSet141,175,0.05)
```

求得 $h=1$,即应放弃原假设而接受 H_1 假设,$p=0.001\,6$ 及 $ci(1)=155.146\,6$, $ci(2)=170.178\,4$。换句话说,可以 99.84% [即(100(1−0.001 6)]地相信 DataSet141 中的数据的平均值不等于 175。

情况 4:

在情况 4 中仍以函数 DataFci 定义的表 14.4 中的数据为例,在 95%的置信度上确定这些样本的平均值是否有显著差异。假设:

$$H_0:\mu_1=\mu_2$$

$$H_1:\mu_1\neq\mu_2$$

现用 ttest2 分析假设的合理性。函数 ttest2 的用法如下:

```
[h,p,ci] = ttest2(x1,x2,alpha)
```

其中 x_1 和 x_2 为数据,$alpha=\alpha$,H_0 成立时 $h=0$,H_1 成立时 $h=1$。对双限置信区间 p 值的计算方法如下:

```
p = 2(1 - tcdf(t0,n - 1))
```

$t0=t_0$ 在表 14.5 中第 4 行第 4 列定义,$ci(1)=l$ 和 $ci(2)=u$ 分别为置信下限和置信上限。对 ttest2 的调用语句如下所述:

```
[x1,x2] = DataFci;
[h,p,ci] = ttest2(x1,x2,0.05)
```

结果为 $h=0$,即不能放弃原假设:$p=0.644\,5$, $ci(1)=-0.755\,0$ 和 $ci(2)=1.185\,5$ 分别为均差的置信下限和置信上限。由 p 值可以看出仅有 35.55% [即 100(1−0.644 5)]的置信度可以相信平均值间有显著性差异,远远低于期望的置信度 95%,因此原假设不能放弃。

情况 7:

在情况 7 中仍以函数 DataFci 定义的表 14.4 所列数据为例。在置信度为 95%时确定这些样本的方差间是否存在显著性差异。假设:

$$H_0:\sigma_1^2=\sigma_2^2$$

$$H_1:\sigma_1^2\neq\sigma_2^2$$

检验统计为:

$$f_0 = \frac{s_1^2}{s_2^2}$$

放弃原假设的依据为：

$$f_0 > f_{\alpha/2, n_1-1, n_2-1} \text{或} f_0 < f_{1-\alpha/2, n_1-1, n_2-1}$$

执行下面的程序清单：

```
[x1,x2] = DataFcı;
L1 = length(x1);
L2 = length(x2);
ratio = var(x1)/var(x2);
if ratio > finv(0.975,L1 - 1,L2 - 1)
  disp('Reject null hypothesis')
  disp(['pValue = 'num2str(2 * (1 - fcdf(ratio,L1 - 1,L2 - 2)))])
elseif ratio < finv(0.025,L1 - 1,L2 - 1)
  disp('Reject null hypothesis')
  disp(['pValue = 'num2str(2 * fcdf(ratio,L1 - 1,L2 - 1))])
else
  disp('Null hypothesis cannot be rejected')
end
```

结果如下：

```
Reject null hypothesis
pValue = 6.5379e-005
```

由上述结果可看出，样本方差间有显著性差异。值得注意的是对一般双限置信区间其 F 检验的 p 值由下述语句求得：

```
p = 2 * (1 - fcdf(r,n1,n2))
```

其中 $r = s_1^2/s_2^2$，$n1 = n_1$ 和 $n2 = n_2$ 为对应的自由度。

14.5 线性回归

14.5.1 一元线性回归

回归分析统计是在分析两变量或更多变量间关系并确定其模型时采用的技术。一元线性回归模型只有一个独立变量。设过程的输入量为 x，输出响应为 y，则其线性模型为：

$$y = \beta_1 x + \beta_0$$

如果有 n 个独立变量 x_i，就应该有 n 个对应的输出响应 y_i，$(i = 1,2,\cdots,n)$，则 y 的估计值由下式确定：

$$\hat{y} = \hat{y}(x) = \hat{\beta}_1 x + \hat{\beta}_0 \qquad x_{min} \leqslant x \leqslant x_{max} \tag{14.26}$$

其中 x_{min}为 x_i 的最小值，x_{max}为 x_i 的最大值，$\hat{\beta}_1$ 和 $\hat{\beta}_0$ 分别为 β_1 和 β_0 的估计值，且由下式给定：

$$\hat{\beta}_1 = \frac{S_{xy}}{S_{xx}}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1\bar{x} \tag{14.27}$$

式中，

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n}x_i \qquad \bar{y} = \frac{1}{n}\sum_{i=1}^{n}y_i$$
$$S_{xx} = \sum_{i=1}^{n}x_i^2 - n\bar{x}^2 \qquad S_{xy} = \sum_{i=1}^{n}x_iy_i - n\overline{xy} \tag{14.28}$$

$\hat{\beta}_1$ 和 $\hat{\beta}_0$ 的值由函数 polyfit 求得(参照第 5.6.1 节)，因此：

```
[c,s] = polyfit(x,y,1)
```

其中 $c(1)=\hat{\beta}_1$，$c(2)=\hat{\beta}_0$，s 为函数 polyconf 所需要的参数，其用法将在下面解释。

对于 $x_{min} \leqslant x \leqslant x_{max}$，$y(x)$估计值的 $100(1-\alpha)\%$ 置信上下限为：

$$\hat{y}(x) - w(x) \leqslant y(x) \leqslant \hat{y}(x) + w(x) \tag{14.29}$$

式中，

$$w(x) = t_{\alpha/2,n-2}\hat{\sigma}\sqrt{1+\frac{1}{n}+\frac{(x-\bar{x})^2}{S_{xx}}}$$
$$\hat{\sigma}^2 = \frac{SS_E}{n-2} \qquad SS_E = S_{yy} - \hat{\beta}_1 S_{xy} \tag{14.30}$$
$$S_{yy} = \sum_{i=1}^{n}y_i^2 - n\bar{y}^2$$

$w(x)$和 $\hat{y}(x)$由函数 polyconf 求得，如下所述：

```
[c,s] = polyfit(x,y,1)
[yhat,w] = polyconf(c,x,s,alpha)
```

其中 $yhat = \hat{y}(x)$，$w = w(x)$，$alpha = \alpha$，向量 x 确定 $yhat$ 和 w 的取值。

检验由式(14.26)给出的回归模型是否恰当的方法是求其残差，即：

$$e_i = y_i - \hat{y}(x_i) \qquad i = 1,2,\cdots,n \tag{14.31}$$

如果 e_i 近似为正态分布，则选用该模型是合适的。

另一种检验模型是否恰当的方法是引入可决系数R^2，R^2 定义为：

$$R^2 = 1 - \frac{SS_E}{S_{yy}} \tag{14.32}$$

值 $100\% \cdot R^2$ 为模型所描述的数据变化率的百分比，该值越接近 100%，模型越有效。R 称为相关系数。

下面介绍这些关系式的应用。

例 14.5　回归分析

有一组数据如表 14.6 所列，该数据由函数 *DataRegress*1 定义，注意表中数据是无序的，由于在图上把这些无序数据与一条直线相连很麻烦，所以先对它们按升序进行排列。然而在 *Polyfit* 和 *Polyconf* 函数中并不需要再次排序。因此：

```
function[x,y] = DataRegress1
```

```
xx = [2.38 2.44 2.70 2.98 3.32 3.12 2.14 2.86 3.50 3.20 2.78 2.70 2.36 2.42...
    2.62 2.80 2.92 3.04 3.26 2.30];
yy = [51.11 50.63 51.82 52.97 54.47 53.33 49.90 51.99 55.81 52.93 52.87 52.36...
     51.38 50.87 51.02 51.29 52.73 52.81 53.59 49.77];
[x,index] = sort(xx);
y = yy(index);
```

上述代码中，*index* 为 x 中每个元素排序前的位置，为保留 x 和 y 的对应关系必须记录 *index* 值。如对 x 和 y 都用 sort 函数排序，则其对应关系将被破坏。

表 14.6 用于一元线性回归分析的数据——*DataRegress1*

x	y	x	y
2.38	51.11	2.78	52.87
2 44	50.63	2.70	52.36
2 70	51.82	2.36	51.38
2.98	52.97	2.42	50.87
3.32	54.47	2.62	51.02
3.12	53.33	2.80	51.29
2.14	49 90	2.92	52.73
2.86	51.99	3.04	52.81
3 50	55.81	3.26	53.59
3.20	52.93	2.30	49.77

下面给出程序清单，该程序求出 $\hat{\beta}_1$ 和 $\hat{\beta}_0$，并在图上画出 $\hat{y}(x)$ 及置信度为 95% 的置信区间。在图中画出各个数据点，然后把这些点与直线相连，再加上恰当注释。另外还计算出可决系数并标于图中。运行结果见图 14.11。

```
[x,y] = DataRegress1;
[c,s] = polyfit(x,y,1);
[yhat,w] = polyconf(c,x,s,.05);
syy = sum(y.^2) - length(x) * mean(y)^2;
sse = syy - c(1) * (sum(x. * y) - length(x) * mean(x) * mean(y));
plot(x,yhat,'k-',x,yhat-w,'k--',x,yhat + w,'k- -',x,y,'ks',[x;x],[yhat;y],'k-')
legend('Regression line','95% confidence interval of y',4)
axis([2 3.6 48 57])
xlabel('x(Input)')
ylabel('y(Response)')
title('Simple linear regression')
text(2.1,56,['Coefficient of determination R^2 = 'num2str(1 - sse/syy,3)])
```

进一步分析残差。首先计算残差值然后用函数 normplot(参见图 14.7)将其画在图中，以观察残差是否服从正态分布。程序为：

```
[x,y] = DataRegress1;
normplot(y-ployval(polyfit(x,y,1),x))
whitebg('white')
```

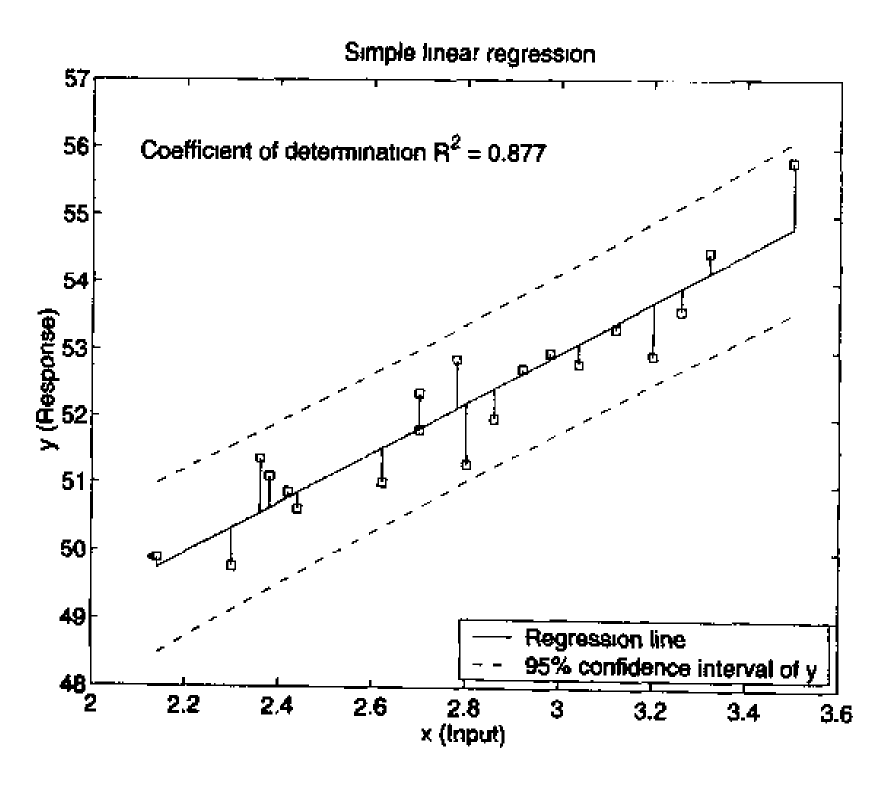


图 14.11　对表 14.6 所列数据进行线性回归分析的结果及 y 的置信上限和下限图

执行结果如图 14.12 所示。由图中看出这些残差点与直线非常接近，由此得出结论，这些残差值非常接近于正态分布，选择的模型是合适的。

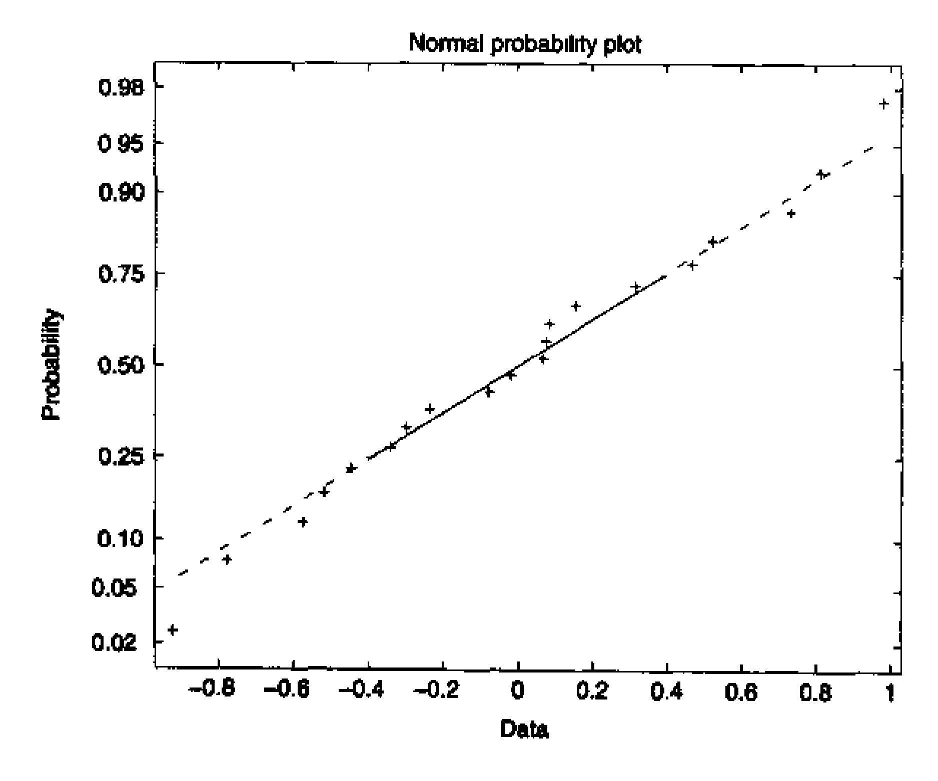


图 14.12　在图 14.11 所示直线附近的残差值的正态累积分布图

14.5.2　多元线性回归

实际应用中，可能影响过程输出量的因素（变量）多于一个，在这种情况下，需要建立多元

回归模型。

在一过程中，有一个输出 y 和 k 个输入 $x_j, j=1,2,\cdots,k$，这个过程可用下述模型描述：

$$y = \beta_o + \sum_{j=1}^{k}\beta_j x_j \tag{14.33}$$

上式为具有 k 个独立变量的多元线性回归模型。$\beta_j, j=0,1,2,\cdots,k$，称做回归系数。更复杂的问题也可用多元线性回归模型描述。例如，在三次多项式中只有一个独立变量 x，如下所述：

$$y = \beta_o + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$$

如令 $x_1 = x, x_2 = x^2, x_3 = x^3$，则得到式(14.33)所示的线性模型，即：

$$y = \beta_o + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$$

而这类模型用函数 polyfit 更易求解。

另一个例子为：

$$y = \beta_o + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \beta_5 x_1 x_2$$

则该方程将与式(14.33)描述 $x_3 = x_1^2, x_4 = x_2^2, x_5 = x_1 x_2$ 的形式一致。

由此看出，任何回归模型当与参数 β_j 呈线性关系时，无论其输出 y 的曲面形状如何，都是一个线性回归模型。

为了估计一些参数值，可以做 n 次重复试验($n > k+1$)。对应于每组数据 $x_{ij}, i=1,2,\cdots,n$ 及 $j=1,2,\cdots,k$，通过实验得到一组输出 y_i。用表格形式表示为：

y	x_1	x_2	$\cdots$	x_k
y_1	x_{11}	x_{12}	$\cdots$	x_{1k}
y_2	x_{21}	x_{22}	$\cdots$	x_{2k}
$\vdots$	$\vdots$	$\vdots$		$\vdots$
y_n	x_{n1}	x_{n2}	$\cdots$	x_{nk}

式(14.33)变为下列形式：

$$y_i = \beta_o + \sum_{j=1}^{k}\beta_j x_{ij} \quad i = 1,2,\cdots,n \tag{14.34}$$

如果数据用矩阵表示，则：

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1k} \\ 1 & x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & & & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nk} \end{bmatrix} \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad \hat{a} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \vdots \\ \hat{\beta}_k \end{bmatrix} \tag{14.35}$$

β_j 的估计值用 $\hat{\beta}_j$ 表示，可由下述矩阵方程求得：

$$\hat{a} = (X'X)^{-1}X'y \tag{14.36}$$

矩阵 X 一般不是方阵。那么，

$$\hat{y}_i = \hat{\beta}_o + \sum_{j=1}^{k}\hat{\beta}_j x_{ij} \quad i = 1,2,\cdots,n \tag{14.37}$$

式中 $\hat{y}_i$ 为 y_i 的估计值。

求得回归系数后，还需计算残差值以分析它们是否服从正态分布，作为验证模型正确性的

一个指标。残差定义如下：

$$\mathbf{e}=\mathbf{y}-\hat{\mathbf{y}}$$

则：

$$e_i = y_i - \hat{y}_i = y_i - \hat{\beta}_0 - \sum_{j=1}^{k}\hat{\beta}_j x_{ij} \quad i = 1,2,\cdots,n \tag{14.38}$$

回归系数 β_j 的置信区间如下：

$$\beta_{Lj} \leqslant \beta_j \leqslant \beta_{Uj} \qquad j=0,1,\cdots,k \tag{14.39}$$

其中，

$$\begin{aligned}\beta_{Lj} &= \hat{\beta}_j - t_{\alpha/2,n-k-1}\hat{\sigma}\sqrt{C_{jj}}\\ \beta_{Uj} &= \hat{\beta}_j + t_{\alpha/2,n-k-1}\hat{\sigma}\sqrt{C_{jj}}\end{aligned} \tag{14.40}$$

和

$$\hat{\sigma}^2 = \frac{\mathbf{y}'\mathbf{y} - \hat{\mathbf{a}}'\mathbf{X}'\mathbf{y}}{n-k-1}$$

$$\mathbf{C} = (\mathbf{X}'\mathbf{X})^{-1} = \begin{bmatrix} C_{00} & C_{01} & \cdots & C_{0k}\\ C_{10} & C_{11} & & \\ \vdots & & \ddots & \\ C_{k0} & & & C_{kk}\end{bmatrix} \tag{14.41}$$

换句话说，

$$\mathrm{var}(\hat{\beta}_j) = \hat{\sigma}^2 C_{jj} \quad j=0,1,\cdots,k$$

为 $\hat{\beta}_j$ 的方差估计值，及

$$\mathrm{covar}(\hat{\beta}_i,\hat{\beta}_j) = \hat{\sigma}^2 C_{ij} \quad i,j=0,1,\cdots,k \quad i\neq j$$

是 $\hat{\beta}_i$ 和 $\hat{\beta}_j$ 的协方差估计值。

多元可决系数 R^2 为：

$$R^2 = 1 - \frac{\mathbf{y}'\mathbf{y} - \hat{\mathbf{a}}'\mathbf{X}'\mathbf{y}}{\mathbf{y}'\mathbf{y} - n\bar{y}^2} \tag{14.42}$$

其中，

$$\bar{y} = \frac{1}{n}\sum_{j=1}^{n} y_j$$

R 为相关系数。

下面进行假设检验，检验是否至少有一个回归变量（$x_i, i=1,2,\cdots,k$）与输出响应 y 之间存在线性关系。假设检验为：

$$H_0: \beta_1 = \beta_2 = \cdots = \beta_k = 0$$

$$H_1: \beta_j \neq 0 \quad (\text{至少有一个 } j \text{ 值})$$

放弃 H_0 假设意味着至少有一个回归变量对统计做出显著贡献。检验统计量定义为：

$$F_0 = \frac{(\hat{\mathbf{a}}'\mathbf{X}'\mathbf{y} - n\bar{y}^2)/k}{(\mathbf{y}'\mathbf{y} - \hat{\mathbf{a}}'\mathbf{X}'\mathbf{y})/(n-k-1)} \quad n > k+1 \tag{14.43}$$

如果，

$$F_0 > f_{\alpha,k,n-k-1}$$

则放弃 H_0。

对上述方程的数值计算可通过语句：

```
beta = regress(y,x)
```

或语句：

```
[beta,betacl,e,ecl,stats] = regress(y,X,alpha)
```

实现。

其中，

beta = $[\hat{\beta}_0\ \hat{\beta}_1 \ldots \hat{\beta}_k]$在方程(14.36)中定义

betacl 为 $((k+1)\times 2)$维数组，存放置信下限 β_L 和置信上限β_U 值，如式(14.40)所定义的那样。阶数由 *beta* 确定。

$e = [e_1\ e_2 \ldots e_n]$为式(14.38)给出的残差；

ecl 为残差的置信区间

stats = $[R^2\ F_0\ P]$，其中，

R^2 由式 (14.42)给出

F_0 由式(14.43)给出

p 为 F_0 对应的 p 值，即：

```
p = 1 - fcdf(F0,k,n-k-1)
```

$y = [y_1\ y_2 \ldots y_n]'$为输出响应列向量

X 等于由式(14.35)定义的 X

alpha = α

下面给出这些公式的具体用法。

例 14.6 多元回归分析

有一组数据如表 14.7 所示，现用下述方程模拟这些数据：

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_1 x_2 + \beta_5 x_1 x_3 + \beta_6 x_2 x_3 + \beta_7 x_1^2 + \beta_8 x_2^2 + \beta_9 x_3^2$$

首先创建函数 DataMultiRegress1，根据式(14.35)生成 X 变量：

```
function[y,X] = DataMultiRegress1
y = 0.22200 0.39500 0.42200 0.43700 0.42800 0.46700 0.44400 0.37800 0.49400 ...
   0.45600 0.45200 0.11200 0.43200 0.10100 0.23200 0.30600 0.09230 0.11600 ...
0.07640 0.43900 0.09440 0.11700 0.07260 0.04120 0.25100 0.00002]';
x1 = [7 3 8 7 8.8 8.1 9.0 8.7 9.3 7.6 10.0 8.4 9.3 7.7 9.8 7.3 8.5 9.5 7.4 7.8 7.7 10.3 ...
    7.8 7.1 7.7 7.4 7.3 7.6]';
x2 = [0.0 0.0 0.7 4.0 0.5 1.5 2.1 5.1 0.0 3.7 3.6 2.8 4.2 2.5 2.0 2.5 2.8 2.8 3.0 1.7 ...
    3.3 3.9 4.3 6.0 2.0 7.8]';
x3 = [0.0 0.3 1.0 0.2 1.0 2.8 1.0 3.4 0.3 4.1 2.0 7.1 2.0 6.8 6.6 5.0 7.8 7.7 8.0 ...
    4.2 8.5 6.6 9.5 10.9 5 2 20 7]';
X = [ones(length(y),1)x1 x2 x3 x1.*x2 x1.*x3 x2.*x3 x1.^2 x2.^2 x3.^2];
```

表 14 7　函数 *DataMultiRegress1* 定义的数据表

y	x_1	x_2	x_3	y	x_1	x_2	x_3
0 222 00	7 3	0 0	0 0	0 101 00	7 3	2 5	6 8
0 395 00	8 7	0 0	0 3	0 232 00	8 5	2 0	6 6
0 422 00	8 8	0 7	1 0	0 306 00	9 5	2 5	5 0
0 437 00	8 1	4 0	0 2	0 092 30	7 4	2 8	7 8
0 428 00	9 0	0 5	1 0	0 116 00	7 8	2 8	7 7
0 467 00	8 7	1 5	2 8	0 076 40	7 7	3 0	8 0
0 444 00	9 3	2 1	1 0	0 439 00	10 3	1 7	4 2
0 378 00	7 6	5 1	3 4	0 094 40	7 8	3 3	8 5
0 494 00	10 0	0 0	0 3	0 117 00	7 1	3 9	6 6
0 456 00	8 4	3 7	4 1	0 072 60	7 7	4 3	9 5
0.452 00	9 3	3 6	2 0	0 041 20	7 4	6 0	10 9
0 112 00	7 7	2 8	7 1	0 251 00	7 3	2 0	5 2
0 432 00	9 8	4 2	2 0	0 000 02	7 6	7 8	20 7

然后计算出系数 β_i 的估计值及其在 95%置信度上的置信区间，在窗口中显示 R^2、F_0 及其 p 值，并在图上绘出残差值，以确定残差是否服从正态分布。下面为程序清单：

```
[y,X] = DataMultiRegress1;
[b,bcl,e,ecl,stat] = regress(y,X,0.05);
lenb = length(b);
disp('Regression coefficients and their confidence limits')
disp([num2str(bcl(:,1))repmat('< = beta(',lenb,1)num2str((0:lenb-1)')...
  repmat(') = ',lenb,1)num2str(b)repmat('< =',lenb,1)num2str(bcl(:,2))])
disp(['Coefficient of determination R^2 = 'num2str(stat(1))])
disp(['Test statistic F0 = 'num2str(stat(2))'and corresponding p-value = '..
        num2str(stat(3))])
normplot(e)
whitebg('white')
```

执行结果在 MATLAB 命令窗口中显示出下列数据。为了观察方便，先用手工在图中列出了这些数据，如图 14.13 所示。从图中可看出除有 5 个点以外其余残差点都落在正态分布直线的附近，因此该模型是合适的。

```
Regression coefficients and their confidence limits
-4.4976     < = beta(0) = -1.7694     < = 0.9589
-0.20282    < = beta(1) = 0.4208      < = 1.0444
-0.054708   < = beta(2) = 0.22245     < = 0.49961
-0.27691    < = beta(3) = -0.128      < = 0.020918
-0.045395   < = beta(4) = -0.019876   < = 0.0056419
-0.0070049  < = beta(5) = 0.0091515   < = 0.025308
-0.012346   < = beta(6) = 0.0025762   < = 0.017499
-0.054932   < = beta(7) = -0.019325   < = 0.016283
-0.032989   < = beta(8) = -0.0074485  < = 0.018092
-0.002231   < = beta(9) = 0.00082397  < = 0.003879
Coefficient of determination R^2 = 0.91695
Test statistic F0 = 19.628 and corresponding p-value = 5.0513e-007
```

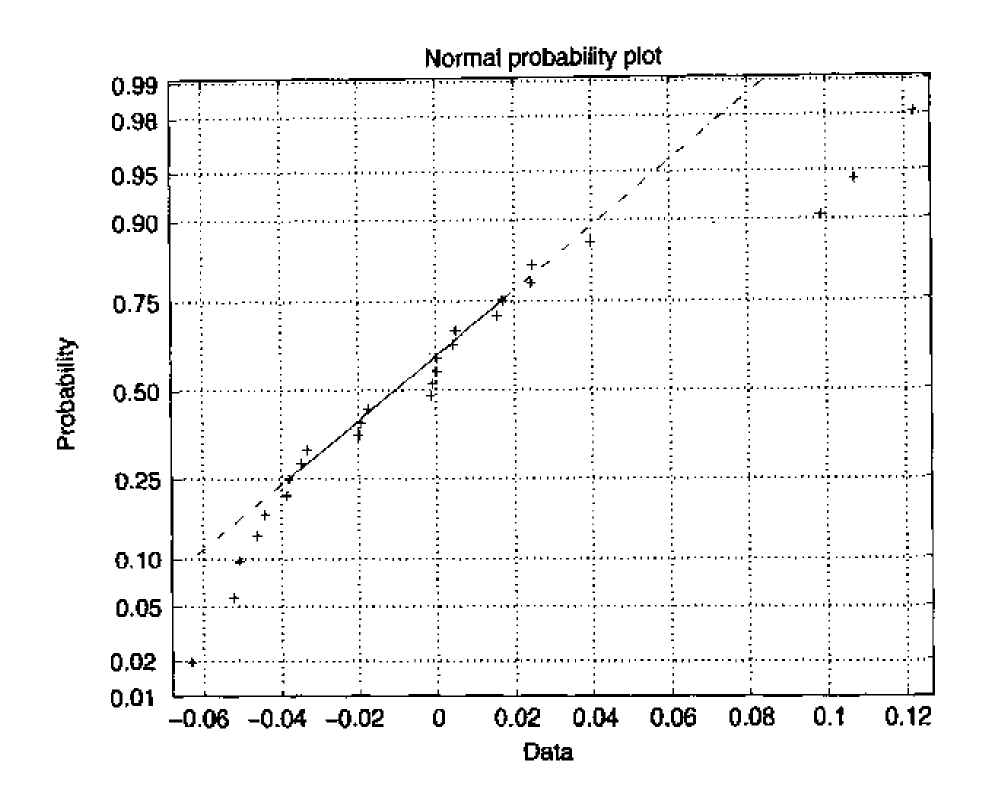


图 14.13 表 14.7 中数据的残差正态累积分布图

14.6 试验设计

14.6.1 单因素实验:方差分析

在单因素实验中,因素用 A 表示。在一次实验过程中,将 A 取不同的水平 A_j,$j=1,2,\cdots,a$,对同样实验重复做 n 次即得到 n 次重复。结果数据填入表 14.8 中。在表 14.8 中,第 1 列数据是观察值 x_{j1},它是对水平 A_j,$(j=1,2,\cdots,a)$随机排序后测得的实验数据。第 2 列观察值 x_{j2},$j=1,2,\cdots,a$,是对水平 A_j 重新随机排序后测得的实验数据,上述过程重复做 n 次可得到 n 组不同数据。做实验时,要保证测出数据 x_{jk} 的独立性。可定义两个独立方差,定义中用到的参数 μ_i 和 s_i^2 见表 14.8 所示。因素 A 的平均方差为:

$$s_A^2 = \frac{n}{a-1}\left(\sum_{i=1}^{a}\mu_i^2 - a\bar{x}^2\right) = \frac{SS_A}{a-1}$$

上式中自由度为 $a-1$,$\bar{x}$ 为总平均值,定义如下:

$$\bar{x} = \frac{1}{an}\sum_{i=1}^{a}\sum_{j=1}^{n}x_{ij}$$

误差的方差为:

$$s_{error}^2 = \frac{1}{a}\sum_{i=1}^{a}s_i^2 = \frac{1}{a(n-1)}\left(\sum_{i=1}^{a}\sum_{j=1}^{n}x_{ij}^2 - an\bar{x}^2\right) = \frac{SS_{error}}{a(n-1)}$$

其中 $a(n-1)$为自由度。

表 14.8　重复 $n>1$ 次的单因素实验表

水平	观察值				均值	方差	残差
A_1	x_{11}	x_{12}	…	x_{1n}	$\mu_1=\frac{1}{n}\sum_{j=1}^{n}x_{1j}$	$s_1^2=\frac{1}{n-1}\sum_{j=1}^{n}(x_{1j}-\mu_1)^2$	$\varepsilon_{1j}=x_{1j}-\mu_1$
A_2	x_{21}	x_{22}		x_{2n}	$\mu_2=\frac{1}{n}\sum_{j=1}^{n}x_{2j}$	$s_2^2=\frac{1}{n-1}\sum_{j=1}^{n}(x_{2j}-\mu_2)^2$	$\varepsilon_{2j}=x_{2j}-\mu_2$
						…	
A_a	x_{a1}	x_{a2}	…	x_{an}	$\mu_a=\frac{1}{n}\sum_{j=1}^{n}x_{aj}$	$s_a^2=\frac{1}{n-1}\sum_{j=1}^{n}(x_{aj}-\mu_a)^2$	$\varepsilon_{aj}=x_{aj}-\mu_a$

下面的等式将两个方差 s_A^2 和 s_{error}^2 联系起来：

$$\begin{aligned}\sum_{i=1}^{a}\sum_{j=1}^{n}(x_{ij}-\bar{x})^2 &= \sum_{i=1}^{a}\sum_{j=1}^{n}[(\mu_i-\bar{x})+(x_{ij}-\mu_i)]^2\\ &= n\sum_{i=1}^{a}(\mu_i-\bar{x})^2+\sum_{i=1}^{a}\sum_{j=1}^{n}(x_{ij}-\mu_i)^2\\ SS_{total} &= SS_A+SS_{error}\\ &= (a-1)s_A^2+a(n-1)s_{error}^2\end{aligned}$$

等式左边被称为总平方和，其自由度为 $an-1$。等式将总方差分为两个独立部分：一部分与因素 A 有关，另一部分与过程数据的误差有关，该误差用 ε_{ij} 表示。

方差分析中，习惯上定义一个均方值参数，用 MS 表示。MS 等于平方和除以自由度，因此对单因素实验为：

$$MS_A=\frac{SS_A}{(a-1)}=s_A^2 \qquad (a>1)$$

$$MS_{error}=\frac{SS_{error}}{a(n-1)}=s_{error}^2 \quad (n>1)$$

实验目的是要确定因素 A 在不同水平上对输出 x_{ij} 是否有显著的影响。为此，定义因素 A 的均方值与独立随机误差均方值的比。由比值大小可知 A 的方差在总方差中是否占有显著位置。因此，检验统计量定义为：

$$F_0=\frac{MS_A}{MS_{error}}$$

建立假设为：

$$H_0:\mu_1=\mu_2=\ldots=\mu_a$$
$$H_1:\mu_j\neq\mu_i \text{ 至少有一个 } j\neq i$$

若，

$$F_0>f_{\alpha,a-1,a(n-1)}$$

则原假设没有被放弃。分析结果通常以表格的形式给出，如表 14.9 所示。

表 14.9　重复 $n>1$ 次的单因素方差分析(ANOVA)表

因素	平方和	自由度	均方值	F_0	p 值
A	SS_A	$a-1$	MS_A	MS_A/MS_{error}	
Error	SS_{error}	$a(n-1)$	MS_{error}		
Total	SS_{total}	$an-1$			

单因素方差分析可由

```
p=anova1(x)
```

实现，其中 p 为 p 值，x 为对表 14.8 的列数据转换后的数据。该函数另外有两个输出：一个为表 14.9 所示的方差分析(ANOVA)表，另一个是对应 a 水平的中位数变化盒状图。下面举例说明单因素实验的方差分析。

例 14.7 单因素方差分析

有一组数据如表 14.10 所示，下面编写程序绘出方差分析 ANOVA 表，显示 p 值，计算残差并分析残差是否服从正态分布。首先创建一函数将数据转换为函数 anova1 能接受的形式，如下所示：

```
function d=DataAnova1
d=[143 141 150 146;..
   152 149 137 143;...
   134 133 132 127; ..
   129 127 132 129;..
   147 148 144 142]';
```

表 14.10 例 14.7 中的数据

水平	观察值			
1	143	141	150	146
2	152	149	137	143
3	134	133	132	127
4	129	127	132	129
5	147	148	144	142

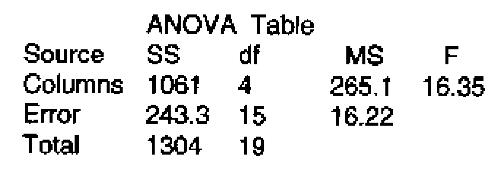
程序如下所示：

```
vv=DataAnova1;
[r,c]=size(vv);
pp=anova1(vv);
meen=mean(vv);
k=0;
for n=1:r
 for m=1:c
  k=k+1;
  e(k)=vv(n,m)-meen(m);
 end
end
disp(['p-value='num2str(pp)])
figure
normplot(e)
whitebg('white')
```

上述代码中在 for 循环内将所有残差放于一阶向量中。函数 figure 用于打开另一图形窗口。因为 anova1 函数自己会打开两个窗口，如不使用 figure 函数，则由函数 anova1 生成的两个图形窗口的其中一个将被重写。下面为 MATLAB 命令窗口显示的结果：

```
p-value=2.414e-005
```

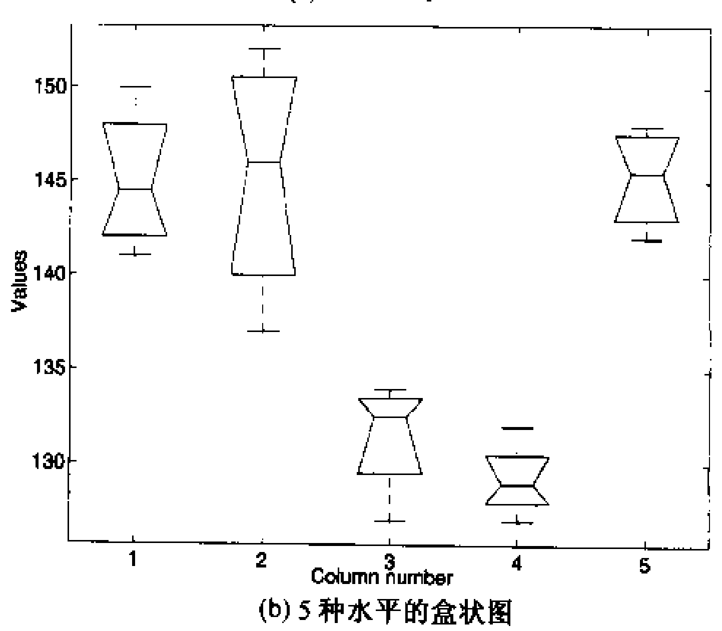
Anova1 函数创建的表格如图 14.14(a)所示,生成的盒状图如图 14.14(b)所示。图 14.14(c)是由 normplot 函数生成的。



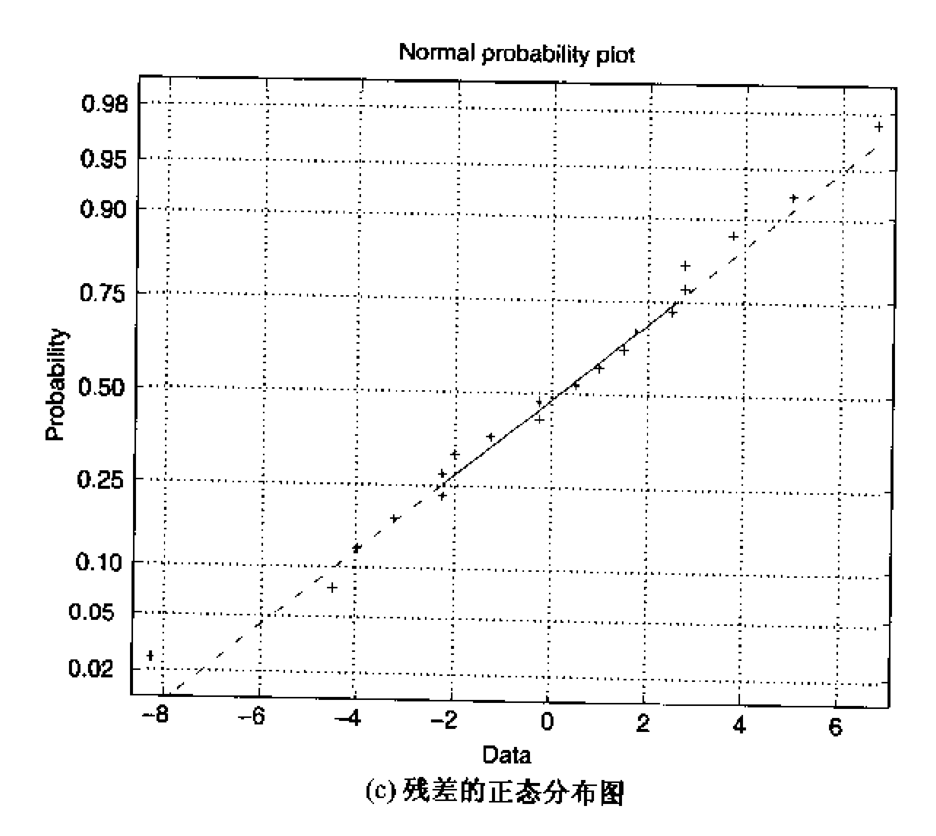
ANOVA Table

Source	SS	df	MS	F
Columns	1061	4	265.1	16.35
Error	243.3	15	16.22	
Total	1304	19		

(a)ANOVA 表



(b) 5 种水平的盒状图



(c) 残差的正态分布图

图 14.14　表 14.10 中数据的方差分析

14.6.2 多因素实验

析因实验 单因素实验结果可延伸用于多因素实验，后者被称做析因实验。析因实验中每次重复要包括每个因素的所有水平的各种不同组合情况。现用两因素实验说明这一点。设实验因素 A 具有 a 个水平，因素 B 具有 b 个水平，实验重复次数为 $n(n>1)$，输出为 x_{ijk}，其中 $i=1,2,\cdots,a;j=1,2,\cdots,b;k=1,2,\cdots,n$。每个因素的不同水平间的间隔不一定相等。实验数据如表 14.11 所示：

表 14.11 两因素析因实验中的数据

		因素 B			
		1	2	…	b
因素 A	1	$y_{111},y_{112},\quad,y_{11n}$	$y_{121},y_{122},\quad,y_{12n}$		$y_{1b1},y_{1b2},\cdots,y_{1bn}$
	2	$y_{211},y_{212},\quad,y_{21n}$	$y_{221},y_{222},\cdots,y_{22n}$		$y_{2b1},y_{2b2},\cdots,y_{2bn}$
	⋮				
	a	$y_{a11},y_{a12},\cdots,y_{a1n}$	$y_{a21},y_{a22},\quad,y_{a2n}$		$y_{ab1},y_{ab2},\cdots,y_{abn}$

首先应从平方和等式分析入手。但在此之前，先引入下列几个不同的平均值的定义：

$$\bar{x}_{ijn}=\frac{1}{n}\sum_{k=1}^{n}x_{ijk}$$

$$\bar{x}_{ibn}=\frac{1}{b}\sum_{j=1}^{b}\bar{x}_{ijn}=\frac{1}{bn}\sum_{j=1}^{b}\sum_{k=1}^{n}x_{ijk}$$

$$\bar{x}_{ajn}=\frac{1}{a}\sum_{i=1}^{a}\bar{x}_{ijn}=\frac{1}{an}\sum_{i=1}^{a}\sum_{k=1}^{n}x_{ijk}$$

总平均值为：

$$\bar{x}=\frac{1}{abn}\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n}x_{ijk}$$

两因素方差分析的总平方和等式为：

$$SS_{total}=\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n}(x_{ijk}-\bar{x})^2=SS_A+SS_B+SS_{AB}+SS_{error}$$

其中，

$$SS_A=\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n}(\bar{x}_{ibn}-\bar{x})^2=bn\sum_{i=1}^{a}\bar{x}_{ibn}^2-abn\bar{x}^2$$

$$SS_B=\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n}(\bar{x}_{ajn}-\bar{x})^2=an\sum_{j=1}^{b}\bar{x}_{ajn}^2-abn\bar{x}^2$$

$$SS_{AB}=n\sum_{i=1}^{a}\sum_{j=1}^{b}(\bar{x}_{ijn}-\bar{x}_{ibn}-\bar{x}_{ajn}+\bar{x})^2$$

和

$$SS_{error}=\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n}(x_{ijk}-\bar{x}_{ijn})^2=\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n}x_{ijk}^2-n\sum_{i=1}^{a}\sum_{j=1}^{b}\bar{x}_{ijn}^2$$

SS_A、SS_B、SS_{AB}、SS_{error}及 SS_{total}的自由度分别为$(a-1)$、$(b-1)$、$(a-1)(b-1)$、$ab(n-1)$及

$abn-1$。平方和 SS_{AB} 为因素 A 和 B 的交互效应的平方和。两因素实验的方差分析表如表14.12所示，表中还给出了均方值的定义。可以看出，方差分析中把两因素的交互效应单独分离出来，通过求 MS_{AB}/MS_{error} 来判断两因素交互效应在给定置信度下统计意义上的显著性。

表 14.12 重复 $n>1$ 次的两因素实验方差分析(ANOVA)表

因素	平方和	自由度	均方值	F_0	$f_{\alpha,z,ab(n-1)}$	p 值
A	SS_A	$a-1$	$MS_A=SS_A/(a-1)$	MS_A/MS_{error}	(f 表，$z=a-1$)	
B	SS_B	$b-1$	$MS_B=SS_B/(b-1)$	MS_B/MS_{error}	(f 表，$z=b-1$)	
AB	SS_{AB}	$(a-1)(b-1)$	$MS_{AB}=SS_{AB}/(a-1)(b-1)$	MS_{AB}/MS_{error}	(f 表，$z=(a-1)(b-1)$)	
Error	SS_{error}	$ab(n-1)$	$MS_{error}=SS_{error}/ab(n-1)$			
Total	SS_{total}	$abn-1$				

两因素析因实验的求解可由语句：

```
p = anova2(y,n)
```

实现。其中 n 为重复次数，p 为 p 值的三元素向量，分别表示两个因素及其交互效应：$p(1)$ = 列因素，$p(2)$ = 行因素，$p(3)$ = 行因素与列因素的交互效应。表 14.11 中的数据用 y 矩阵表示如下：

$$y=\begin{bmatrix} y_{111} & y_{121} & \cdots & y_{1b1} \\ y_{112} & y_{122} & & y_{1b2} \\ \vdots & & & \\ y_{11n} & y_{12n} & & y_{1bn} \\ y_{211} & y_{221} & & y_{2b1} \\ y_{212} & y_{222} & & y_{2b2} \\ \vdots & & & \\ y_{21n} & y_{22n} & & y_{2bn} \\ \vdots & & & \\ y_{a11} & y_{a21} & & y_{ab1} \\ y_{a12} & y_{a22} & & y_{ab2} \\ \vdots & & & \\ y_{a1n} & y_{a2n} & & y_{abn} \end{bmatrix}$$

下面举例说明这些关系。

例 14.8 两因素方差分析

一组数据如表 14.13 所示，下面创建函数存放表中数据：

```
function d = DataAnova2
dc1 = [[130 155 74 180]';[150 188 159 126]';[138 110 168 160]'];
dc2 = [[34 40 80 75]';[136 122 106 115]';[174 120 150 139]'];
dc3 = [[20 70 82 58]';[25 70 58 45]';[96 104 82 60]'];
d = [dc1 dc2 dc3];
```

程序为：

```
pvalue = anova2(DataAnova2,4);
disp(['p value of column = 'num2str(pvalue(1))])
disp(['p value of row = 'num2str(pvalue(2))])
disp(['p value of column and row = 'num2str(pvalue(3))])
```

执行结果在 MATLAB 命令窗口中显示：

```
p value of column = 1.9086e-007
p value of row = 0.0019761
p value of column and row = 0.018611
```

在图形窗口中显示下列表格：

ANOVA Table

Source	SS	df	MS	F
Columns	3.912e+004	2	1.956e+004	28.97
Rows	1.068e+004	2	5342	7.911
Interaction	9614	4	2403	3.56
Error	1.823e+004	27	675.2	
Total	7.765e+004	35		

由 p 值看出，因素 A 和因素 B 具有置信度高于 99.8% 的统计意义上的显著性，其交互效应有置信度 98% 的统计意义上的显著性。

表 14.13 例14.8 的数据——DataAnova2

		因素 B		
		1	2	3
因素 A	1	130,155,74,180	34,40,80,75	20,70,82,58
	2	150,188,159,126	136,122,106,115	25,70,58,45
	3	138,110,168,160	174,120,150,139	96,104,82,60

2^k 析因实验 上节描述的析因实验中如果包含 k 个因素且每个因素只有两个水平，则此类析因实验被称做 2^k 析因实验。2^k 析因实验假定在每个因素的两个水平之间存在着线性关系，这种假定给怎样进行实验及分析实验带来了一些简化。

习惯上某因素的高水平用“1”或“+”表示，其低水平用“0”或“-”表示。表 14.14 中列出 k 个因素（$k=2,3,4$），取不同水平的 2^k 个因素组合也在表 14.14 中给出，一次执行顺序即为一次重复。下面介绍表的用法：对于 2^2（$k=2$）析因实验，只有标为 A 和 B 的两列及前 4 行数据有效（$m=1,\cdots,4$）。两个因素的四种组合按随机顺序组成，标为 2^2 的一列即为一种随机组合。首先执行第 2 行的组合：A 为高水平（A_{high}），B 为低水平（B_{low}），其输出值为 $y_{2,1}$，然后是第 4 行的组合，其中 A 和 B 都为高水平（分别为 A_{high} 和 B_{high}），其输出值为 $y_{4,1}$。其余两种组合都做完后，就完成了实验的一次重复。如果重新获得的随机顺序不同于 2^2 列中的顺序，则在该顺序下做四种不同组合，得到第 2 次重复的输出。$k=3$ 时，因素为 A,B,C，表中前 8 行有效；$k=4$ 时，因素有 A,B,C,D，表中所有 16 行数据均有效。表中标为 2^3 和 2^4 的列分别是上述两种情况下的一种随机执行顺序。

表 14.14 不同次序下每个因素在不同水平下的 2^2、2^3 及 2^4 析因实验

运行序列号	因数及其水平				数据($y_{m,j}$)		运行次序号 *		
m	A	B	C	D	$j=1$	$j=2$	2^2	2^3	2^4
1	-	-	-	-	y_{11}	y_{12}	3	5	6
2	+	-	-	-	y_{21}	y_{22}	1	7	11
3	-	+	-	-	y_{31}	y_{32}	4	8	14
4	+	+	-	-	y_{41}	y_{42}	2	4	5
5	-	-	+	-	y_{51}	y_{52}		2	13
6	+	-	+	-	y_{61}	y_{62}		1	2
7	-	+	+	-	y_{71}	y_{72}		3	16
8	+	+	+	-	y_{81}	y_{82}		6	15
9	-	-	-	+	y_{91}	y_{92}			9
10	+	-	-	+	$y_{10\,1}$	$y_{10\,2}$			7
11	-	+	-	+	$y_{11\,1}$	$y_{11\,2}$			10
12	+	+	-	+	$y_{12\,1}$	$y_{12\,2}$			3
13	-	-	+	+	$y_{13\,1}$	$y_{13\,2}$			8
14	+	-	+	+	$y_{14\,1}$	$y_{14\,2}$			4
15	-	+	+	+	$y_{15,1}$	$y_{15\,2}$			1
16	+	+	+	+	$y_{16\,1}$	$y_{16\,2}$			12

* 只是 $j=1$ 的一种随机顺序,对 $j=2$ 将生成一种新的随机顺序,依此类推。

实验数据收集完整后,进行下述分析,设实验重复次数大于 1。观察表 14.15 中所列数据(参考表 4.2),列中符号"+"和"-"分别代表 +1 和 -1,列 A,B,C,D 与表 14.14 中相同,表 14.14 中的符号"+"和"-"也分别代表 +1 和 -1。表示所有交互效应的列值由相应元素列值相乘得到。即标为 ABC 的列的符号值由列 A、列 B 和列 C 中的符号值相乘得到。例如,第 7 行($m=7$)的 $A=-1, B=+1, C=+1$,因此第 7 行标为 ABC 的列值为 $-1[=(-1)(+1)(+1)]$。而且对 2^2 实验只用到表中前 3 列和前 4 行($m=1,2,\cdots4$);对于 2^3 实验,用到表中前 7 列和前 8 行($m=1,2,\cdots,8$)数据;对于 2^4 实验,用到表中前 15 列和 16 行($m=1,2,\cdots,16$)数据。表 14.15 中符号的计算方法见第 4.3.1 节。

表 14.15 在 2^2、2^3 及 2^4 实验中用于求平方和与均方值的不同量的定义

因素及其交互效应(λ)①															数据②			
A	B	AB	C	AC	BC	ABC	D	AD	BD	ABD	CD	ACD	BCD	ABCD	$j=1$	$j=n$	$S_m^{\#}$	m
-	-	+	-	+	+	-	-	+	+	-	+	-	-	+	$y_{1,1}$	$y_{1\,n}$	S_1	1
+	-	-	-	-	+	+	-	-	+	+	+	+	-	-	$y_{2\,1}$	$y_{2\,n}$	S_2	2
-	+	-	-	+	-	+	-	+	-	+	+	-	+	-	$y_{3\,1}$	$y_{3,n}$	S_3	3

(续表)

因素及其交互效应(λ)①															数据②			
														A				
+	+	+	−	−	−	−	−	−	−	−	+	+	+	+	$y_{4\,1}$	$y_{4\,n}$	S_4	4
−	−	+	+	−	−	+	−	+	+	−	−	+	+	−	$y_{5\,1}$	$y_{5\,n}$	S_5	5
+	−	−	+	+	−	−	−	−	+	+	−	−	+	+	$y_{6\,1}$	$y_{6\,n}$	S_6	6
−	+	−	+	−	+	−	−	+	−	+	−	+	−	+	$y_{7\,1}$	$y_{7\,n}$	S_7	7
+	+	+	+	+	+	+	−	−	−	−	−	−	−	−	$y_{8\,1}$	$y_{8\,n}$	S_8	8
−	−	+	−	+	+	−	+	+	−	+	−	+	+	−	$y_{9\,1}$	$y_{9\,n}$	S_9	9
+	−	−	−	−	+	+	+	−	−	−	−	−	+	+	$y_{10\,1}$	$y_{10\,n}$	S_{10}	10
−	+	−	−	+	−	+	+	+	+	−	−	+	−	+	$y_{11\,1}$	$y_{11,n}$	S_{11}	11
+	+	+	−	−	−	−	+	−	+	+	−	−	−	−	$y_{12\,1}$	$y_{12\,n}$	S_{12}	12
−	−	+	+	−	−	+	+	+	−	+	+	−	−	+	$y_{13\,1}$	$y_{13\,n}$	S_{13}	13
+	−	−	+	+	−	−	+	−	−	−	+	+	−	−	$y_{14\,1}$	$y_{14\,n}$	S_{14}	14
−	+	−	+	−	+	−	+	+	+	−	+	−	+	−	$y_{15\,1}$	$y_{15\,n}$	S_{15}	15
+	+	+	+	+	+	+	+	−	+	+	+	+	+	+	$y_{16\,1}$	$y_{16\,n}$	S_{16}	16

① "+"和"−"号分别表示+1和−1,它们也指因素的高水平和低水平。

② 数据的获取方法与表 14 14 一致。

$$^{\#}S_m = \sum_{j=1}^{n} y_{m,j}$$

具有 $n(n>1)$次重复 k 个因素的实验，其数据平方和为：

$$SS_{total} = \sum_{j=1}^{n}\sum_{m=1}^{2^k} y_{m,j}^2 - 2^k n\bar{y}^2$$

$$SS_{error} = SS_{total} - \sum_{\lambda} SS_{\lambda}$$

$$SS_{\lambda} = \frac{C_{\lambda}^2}{n2^k} \qquad \lambda = \mathrm{A,B,AB},\cdots$$

其中，

$$C_{\lambda} = \sum_{m=1}^{2^k} S_m \times (\lambda\text{ 列 } m \text{ 行中的符号}) \qquad \lambda = \mathrm{A,B,AB},\cdots$$

$$\bar{y} = \frac{1}{n2^k}\sum_{m=1}^{2^k} S_m$$

S_m 在表 14.15 中定义。

主因素及其交互效应产生的效果平均值由下式决定：

$$\mathrm{Effect}_{\lambda} = \frac{C_{\lambda}}{n2^{k-1}} \qquad \lambda = \mathrm{A,B,AB},\cdots$$

式中 $\mathrm{Effect}_{\lambda}$ 被称为 λ 的效果。从表 14.15 中可以看出，$k=2$ 时，有 3 个 λ：A、B 和 AB；$k=3$ 时有 7 个 λ：A、B、C、AB、AC、BC 及 ABC；$k=4$ 时有 15 个 λ：A、B、C、D、AB、AC、BC、AD、BD、

CD、ABC、ABD、ACD、BCD 及 $ABCD$。

由于单因素及其交互效应的自由度为 1,所以其均方值为:

$$MS_{\lambda} = SS_{\lambda}$$

误差的均方值为:

$$MS_{error} = \frac{SS_{error}}{2^k(n-1)} \qquad n > 1$$

每一因素及其交互作用的检验统计定义为:

$$F_{\lambda} = \frac{MS_{\lambda}}{MS_{error}} = \frac{MS_{\lambda}}{SS_{error}/[2^k(n-1)]} \qquad \lambda = \mathrm{A,B,AB},\cdots \quad n > 1$$

2^k 析因实验的方差分析表如表 14.16 所示。

表 14.16 重复 $n>1$ 次的 2^k 析因实验的方差分析表

因素	平方和	自由度	均方值	F_{λ}	$f_{\alpha,1,(n-1)2^k}$
A	SS_A	1	MS_A	MS_A/MS_{error}	(f 表中的值)
B	SS_B	1	MS_B	MS_B/MS_{error}	(f 表中的值)
C	SS_C	1	MS_C	MS_C/MS_{error}	(f 表中的值)
⋮					
AB	SS_{AB}	1	MS_{AB}	MS_{AB}/MS_{error}	(f 表中的值)
AC	SS_{AC}	1	MS_{AC}	MS_{AC}/MS_{error}	(f 表中的值)
BC	SS_{BC}	1	MS_{BC}	MS_{BC}/MS_{error}	(f 表中的值)
⋮					
ABC	SS_{ABC}	1	MS_{ABC}	MS_{ABC}/MS_{error}	(f 表中的值)
⋮					
Error	SS_{error}	$2^k(n-1)$	MS_{error}		
Total	SS_{total}	$n2^k-1$			

2^k 析因设计的方差分析表可直接用于求多元回归模型,该模型对其过程的输出量进行估计。该过程的输出为主因素及其交互效应统计意义上显著性的函数。首先引入编码变量 x_{β}:

$$x_{\beta} = \frac{2\beta - \beta_{low} - \beta_{high}}{\beta_{high} - \beta_{low}}$$

其中 β 为主变量,即 $\beta = A, B, C, \cdots$。若 $\beta = A$,则当 $\beta = A_{high}$时,$x_A = +1$;而当 $\beta = A_{low}$时,$x_A = -1$。若 $A_{low} \leqslant A \leqslant A_{high}$,则 $-1 \leqslant x_A \leqslant +1$。

平均输出量 y_{avg}的估计式为:

$$y_{avg} = \bar{y} + 0.5\left[\sum_{\lambda}\mathrm{Effect}_{\lambda}x_{\lambda} + \sum_{\lambda}\sum_{\beta}\mathrm{Effect}_{\lambda\beta}x_{\lambda}x_{\beta} + \sum_{\lambda}\sum_{\beta}\sum_{\gamma}\mathrm{Effect}_{\lambda\beta\gamma}x_{\lambda}x_{\beta}x_{\gamma}\cdots\right]$$

式中 $\lambda,\beta,\gamma,\cdots$为 $A,B,C,\cdots$的值,且仅与那些表示显著性因素和显著交互效应的下标组合一致,$x_{\alpha}(-1 \leqslant x_{\alpha} \leqslant +1)$为编码值。

下面举例说明这些关系。

例 14.9 2^4 析因实验的分析

对表 14.17 中的数据建立方差分析表,表中数据为重复两次的 2^4 析因实验数据。在方差分析表中将包含效果。实验执行序号与表 14.14 中的一致。

首先创建函数 FactorialData,定义表中数据:

```
function dat = FactonalData
dat1 = [159 168 158 166 175 179 173 179 164 187 163 185 168 197 170 194]';
dat2 = [163 175 163 168 178 183 168 182 159 189 159 191 174 199 174 198]';
dat = [dat1 dat2];
```

表 14.17 重复两次的 2^4 析因实验数据

运行序号	数据($y_{m,j}$)	
m	$j=1$	$j=2$
1	159	163
2	168	175
3	158	163
4	166	168
5	175	178
6	179	183
7	173	168
8	179	182
9	164	159
10	187	189
11	163	159
12	185	191
13	168	174
14	197	199
15	170	174
16	194	198

* 运行序号与表 14.14 的水平组合顺序一致。

然后将第 4.3.1 节中的程序转换为一函数,名为 FactorialSigns,该函数确定出表 14.15 中的符号,如下所示:

```
function s = FactonalSigns(k)
s = ones(2^k,2^k-1);
for r = 1:2:2^k
 s(r,1) = -1;
end
for c = 2:k
 e = 2^(c-1);
 for r = 1:e
  s(r,e) = -1;
 end
 for r = e+1:2^(k)
  s(r,2^(c-2)) = s(r-e,2^(c-2));
 end
end
```

```
for m=2:k
e=2^(m-1);
for j=1:e-1
  s(:,e+j)=s(:,j).*s(:,e);
 end
end
```

生成方差分析表的程序如下:

```
tag=str2mat('A','B','AB','C','AC','BC','ABC','D','AD','BD','ABD',   )
              'CD','ACD','BCD','ABCD');
K=4;n=2;
fdata=FactorialData;
s=FactorialSigns(k);
Sm=sum(fdata')';
yBar=sum(Sm)/n2^k;
SStotal=sum(sum(fdata.^2))-yBar^2*n*2^k;
for nn=1:2^k-1
 Clambda=sum(s(:,nn).*Sm);
 SSlambda(nn)=Clambda^2/2^(k+1);
 EffectLambda(nn)=Clambda./2^k;
end
SSerror=SStotal-sum(SSlambda);
MSerror=SSerror/2^k;
f0=SSlambda/MSerror;
pValue=1-fcdf(f0,1,2^k);
disp('Factor      SS      MS      Effect      f-lambda      p-value')
disp([tag repmat('  ',15,1)num2str(SSlambda',6)repmat('  ',15,1)...
     num2str(SSlambda',6)repmat('  ',15,1)num2str(EffectLambda',6)...
     repmat('  ',15,1)num2str(f0',6)repmat('  ',15,1)num2str(pValue',6)])
disp(['SSerror 'num2str(SSerror,6)' 'num2str(MSerror,6)])
disp(['SStotal 'num2str(SStotal,6)])
disp(['yBar='num2str(yBar,6)])
```

其执行结果在 MATLAB 命令窗口中显示下列数据。为清晰起见,手工对数据重新对齐。

Factor	SS	MS	Effect	f-lambda	p-value
A	2312	2312	17	241.778	4.45067e-011
B	21.125	21.125	-1.625	2.20915	0.156633
AB	0.125	0.125	-0.125	0.0130719	0.910397
C	946.125	946.125	10.875	98.9412	2.95785e-008
AC	3.125	3.125	-0.625	0.326797	0.575495
BC	0.5	0.5	-0.25	0.0522876	0.822026
ABC	4.5	4.5	0.75	0.470588	0.502537
D	561.125	561.125	8.375	58.6797	9.69219e-007
AD	666.125	666.125	9.125	69.6601	3.18663e-007
BD	12.5	12.5	1.25	1.30719	0.269723
ABD	2	2	-0.5	0.20915	0.653583
CD	12.5	12.5	-1.25	1.30719	0.269723
ACD	0	0	0	0 1	
BCD	0.125	0.125	0.125	0.0130719	0.910397
ABCD	21.125	21.125	-1.625	2.20915	0.156633
SSerror	153	9.5625			

```
SStotal   4716
yBar =    175.25
```

由此看出,因素 A,C,D 及交互效应 AD 在大大高于95%的置信度上显著影响了过程的输出量。事实上,对这四个因素平方和求和,结果为4 485,因此这些量平方和的不显著贡献为78 = 4 716 − 4 485 − 153,占总平方和的1.65%。

下面将用上述结果求得置信度高于95%的回归方程:

$$y_{avg} = 175.25 + 8.50x_A + 5.44x_C + 4.10x_D + 4.56x_Ax_D$$

残差为测量值 $y_{m,j}$ 与 y_{avg} 间的差,其中 $y_{m,j}$ 与 y_{avg} 为表14.14中 x_γ 编码值的 2^4 种组合所对应的数据。下面编程计算残差,并用 normplot 函数在图中标出残差点。

```
fdata = FactorialData;
s = FactorialSigns(4);
yAvg = 175.25 + 8.5 * s(:,1) + 5.44 * s(:,3) + 4.1 * s(:,4) + 4.56 * s(:,1).* s(:,4);
normplot([fdata(:,1)-yAvg;fdata(:,2)-yAvg])
whitebg('white')
```

执行结果见图14.15,由图中看出得到的残差值是可以接受的。

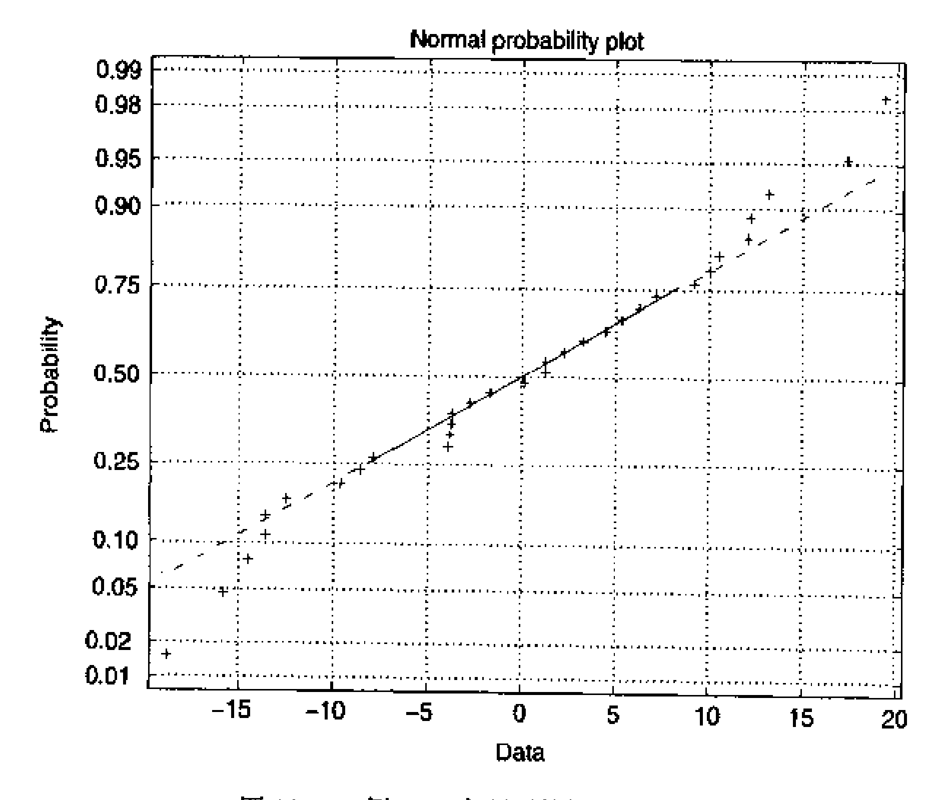


图14.15 例14.9中所用数据的残差图

练习

14.1 有一组数据如表14.18所示,下面假定两种情况:(1)表中所有数据为一集合,用 S_0 表示;(2)将列分为5组,每组数据单独为集合,用 S_j 表示,$j = 1,2,\cdots,5$。

(a) 求 S_0 的调和平均值,并将它与算术平均值及几何平均值进行比较。调和平均

值定义为 $\bar{x}_h$：

$$\frac{1}{\bar{x}_h} = \sum_{i=1}^{n} \frac{1}{x_i}$$

(b) 求 6 组数据各自的平均值及标准偏差 $S_j, j = 0,1,2,\cdots,5$。

(c) 画出数据集 $S_j(j = 1,2,\cdots,5)$ 的直方图。

(d) 求出 S_0 平均值与每个 $S_j(j = 1,2,\cdots,5)$ 平均值之差的置信区间。置信度为 95%，假定标准偏差未知，但均相等(参考表 14.3 和表 14.5 中的情况 4)。每组数据的 p 值为多少？数据组 S_j，其平均值与 S_0 平均值是否存在显著性差异？定性分析所求结果与(c)中图形是否一致？

表 14.18　练习 14.1 中的数据

1		2		3		4		5	
1 115	1 567	1 223	1 782	1 055	798	1 016	2 100	910	1 501
1 310	1 883	375	1 522	1 764	1 020	1 102	1 594	1 730	1 238
1 540	1 203	2 265	1 792	1 330	865	1 605	2 023	1 102	990
1 502	1 270	1 910	1 000	1 608	2 130	706	1 315	1 578	1 468
1 258	1 015	1 018	1 820	1 535	1 421	2 215	1 269	758	1 512
1 315	845	1 452	1 940	1 781	1 109	785	1 260	1 416	1 750
1 085	1 674	1 890	1 120	1 750	1 481	885	1 888	1 560	1 642

表 14.19　练习 14.4 中的数据

过程 #1		过程 #2	
88 4	89 0	92 6	93 2
93 2	90 5	93 2	91 7
87 4	90 8	89 2	91 5
94 3	93 1	94 8	92.0
93 0	92 8	93 3	90 7
94 3	91 9	94 0	93 8

14.2 (a) 一公司的热线帮助系统在工作时间内平均每分钟收到 5 个电话。求下列情况下，该公司能收到电话的概率为多少？(i)每分钟 8 个电话，(ii)每分钟 2 个电话。

(b) 电话系统每分钟能处理 10 个电话，如果多于 10 个，则打电话者将收到占线信号，求收到占线信号的概率为多少？

[答案：(a)(i)0.065 278，(ii)0.084 224，(b)0.013 695。]

14.3 一构件能承受负载 L_0 的概率为 0.7，如果用 15 个此类构件，问至少有 12 个能承受负载 L_0 的概率为多少？

[答案：0.296 87。]

14.4 根据 Taguchi 的定义，过程的平均损失系数与下式成比例：

$$L_{avg} = s^2 + (\bar{x} - \tau)^2$$

式中为目标的平均值。换句话说,如果有两个过程,那么其中平均值最接近于 τ,方差最小的过程具有最低损失系数。对于表 14.19 所列数据,$\tau = 92.0$ 时,确定哪个过程具有最低平均损失系数。

[答案:$L_1 = 5.5904$, $L_2 = 2.6936$。]

14.5 生产商发现有一种产品,其中的 20% 重量不足,这种产品一箱装有 24 件。如假定每件产品的重量与另一件产品的重量无关,则可以应用二项分布解决下列问题:

(a) 求装入箱子内重量不足产品的期望件数及其方差?

(b) 在一箱内重量不足的产品不多于两件的概率为多少?

(c) 箱内没有重量不足产品的概率为多少?

(d) 在同一图上画出概率质量函数和累积分布函数随箱内重量不足产品数变化的曲线。

[答案:(a)期望值 = 4.8,方差 = 3.84;(b)0.11452;(c)0.0047224。]

14.6 X^2 统计用于检验拟合优度。假定有 k 类(k 个单元),每一类的期望值用 $\hat{p}_i$ 表示,$i = 1,2,\cdots,k$。在一实验过程中,分布于每类的观察值(发生值)个数为 x_i, $i = 1,2,\cdots,k$,则 X^2 检验统计为:

$$X^2 = \sum_{i=1}^{k}(x_i - e_i)^2/e_i$$

其中 $e_i = n\hat{p}_i$,且:

$$n = \sum_{i=1}^{k} x_i = \sum_{i=1}^{k} e_i$$

如果任一类中的 $e_i < 5$,则必须将对应的 e_i 和 x_i 分别与其相邻的 e_i 和 x_i 组合直到 $e_i \geqslant 5$。

现用检验统计对下述假设做出检验:

$$H_0: p_i = \hat{p}_i \quad i = 1,\cdots,k$$

式中 $p_i = x_i/n$,如果:

$$X^2 \leqslant \chi^2_{\alpha,k-1}$$

则接受假设 H_0。实际中,α 未给定,而 p 值是给定的。p 值越接近于 1,越易相信观察得出的类发生值 x_i 与期望发生值 e_i 接近。

(a) 假定一种设备的故障发生概率如下所示:发生机械故障的概率为 0.60($\hat{p} = 0.6$),发生电故障的概率为 0.25($\hat{p}_2 = 0.25$),由操作引起的故障概率为 0.15($\hat{p}_3 = 0.15$)。在记录的 55($n = 55$)起设备故障中,有 32 起为机械故障($x_1 = 32$),14 起为电子故障($x_2 = 14$),9 起为操作故障($x_3 = 9$)。用上述方法判断值为 $\hat{p}_i$ 这台设备故障的期望值这一说法的合理性。

[答案:p 值 = 0.94979,因此是合理的。]

(b) 概率 $\hat{p}_i$ 也可由统计模型得出。例如,假定有一基于 λ(假想值的)泊松分布,且:

$$e_i = n\frac{e^{-\lambda}\lambda^{i-1}}{(i-1)!} \quad i=1,2,\cdots,k+1$$

对 $i>k+1$，为：

$$e_i = n\left(1-\sum_{i=1}^{k+1}\frac{e^{-\lambda}\lambda^{i-1}}{(i-1)!}\right) \quad i>k+1$$

如果在容量为85的样本中，每件产品的故障次数如表14.20所示，则判断产品的故障次数服从 $\lambda=3$ 的泊松分布这一结论是否不合理。这里 $k=8$，而且需要对某些 e_i 和 x_i 进行重组。可以利用练习4.4的结果。

[答案：x_i = [17 20 25 14 6 3]，e_i = [16.927 6,19.043 6,19.043 6,14.282 7,8.569 6,7.133 0]，p 值 = 0.405 92 所以结论有些合理。]

表14.20 练习14.6中的数据

i	故障次数($i-1$)	发生次数 x_i
1	0	3
2	1	14
3	2	20
4	3	25
5	4	14
6	5	6
7	6	2
8	7	0
9	8	1
10	≥9	0

14.7 为确定能否使用表14.5中的情况4或情况5，首先要对情况7中表示的方差比值进行 F 检验。如果方差之间显著不同，则可用情况5分析；否则用情况4分析。要求编写一段程序确定表14.21中两组数据平均值之间是否存在差异，根据程序运行结果确定这一点，并形成一盒状图，从图中定性分析上述结论是否正确。

[答案：对方差比值做 F 检验，结果 $p=0.470\,92$，因此方差之间没有差异。对平均值之差做 t 检验，结果 $p=0.000\,934\,2$，因此平均值间有差异。]

表14.21 练习14.7的数据

组1		组2	
88	81	76	79
79	83	83	85
84	90	78	76
89	87	80	80
81	78	84	82
83	80	86	78
82	87	77	78
79	85	75	77
82	80	81	81
85	88	78	80

14.8 过程能力比例(PCR,Process Capability Ratio)是对某一过程是否满足由下限 LSL 和上限 USL 描述的一种度量。对于集中过程 PCR 定义为：

$$PCR = \frac{USL - LSL}{6\hat{\sigma}}$$

对于一非集中过程,PCR 定义为：

$$PCR_k = \min\left[\frac{USL - \bar{x}}{3\hat{\sigma}}, \frac{\bar{x} - LSL}{3\hat{\sigma}}\right]$$

为过程标准偏差的估计值,$\hat{\sigma}$ 为它和 $\bar{x}$ 平均值的估计值。当 $PCR > 1$ 时,几乎没有生产出坏部件或不合格部件;当 $PCR = 1$ 时,有 0.27%(每百万有 2700 个)不合格部件;当 $PCR < 1$ 时,生产出大量不合格部件。$100/PCR$ 为过程用到的定义宽度的百分值。当 $PCR = PCR_k$ 时,过程为集中的。

不合格部件数为 Np,其中 N 为生产部件的总量,且满足下式：

$$p = 1 - \Phi\left(\frac{USL - \bar{x}}{\hat{\sigma}}\right) + \Phi\left(\frac{LSL - \bar{x}}{\hat{\sigma}}\right)$$

式中 Φ 由式(14.16)定义,同时可参考图 14.5c 和式(14.18)。

对表 14.22 中所列数据,用 MATLAB 的 capable 函数求出 p、PCR 和 PCR_k,已知 $LSL = 2.560$, $USL = 2.565$。问该过程是否为集中的?

[答案:$p = 1.535\,1e-004$, $PCR = 1.310\,3$, $PCR_k = 1.209\,9$。]

表 14.22 练习 14.8 中的数据

2 562 9	2 563 0
2 563 0	2 562 8
2 562 8	2 562 3
2 563 4	2 563 1
2 56i 9	2 563 5
2 56l 3	2 562 3

14.9 一部件的可靠率 $R(t)$ 为其在一段时间 t 内运行不失败的概率。如果部件寿命的概率分布函数为 $f(t)$,则其累积分布为：

$$F(t) = P(T \leqslant t) = \int_{-\infty}^{t} f(u)\mathrm{d}u = \int_{0}^{t} f(u)\mathrm{d}u$$

即运行直到失败时的概率。因此：

$$R(t) = 1 - F(t)$$

故障率函数 $h(t)$ 指部件在 t 时刻前未失败在 t 时刻突然失败的偶然性,由下式求出：

$$h(t) = \frac{f(t)}{R(t)} = \frac{f(t)}{1 - F(t)}$$

(a) $f(t)$ 为一指数分布,且满足：

$$f(t) = \frac{1}{\mu}e^{-t/\mu}$$

要求 $\mu=1$ 用 exppdf 函数和 expcdf 函数在同一图上画出故障率函数和可靠率函数。

(b) 如果 $f(t)$ 服从威布尔分布，且 $\alpha=1, \beta=0.5$、1、2 和 4，画出故障率函数和可靠率函数曲线。用 subplot 创建一含 4 个图形的 (2×2) 维数组。每一图形都包含对应一个 β 值的一对曲线。

14.10 (a) 用下列模型对表 14.23 所列数据求 β_j，并说明该模型能很好地拟合表中数据。

$$y=\beta_0+\beta_1x_1+\beta_2x_2+\beta_3x_1^2+\beta_4x_2^2+\beta_5x_1x_2$$

(b) 用求得的 β_j 画出曲面图，并用 contour 函数画出该曲面在 (x_1,x_2) 平面上的等高线投影。

(c) 求出曲面的极大值坐标。

[答案：$x_1=18.7635, x_2=38.0156$。]

表 14.23　练习 14.10 中的数据

y	x_1	x_2
144	18	52
142	24	40
124	12	40
64	30	48
96	30	32
74	26	56
136	26	24
54	22	64
92	22	16
96	14	64
92	10	56
82	10	24
76	6	48
68	6	32

14.11 对数分布的累积分布函数如下所述：

$$F(t)=\Phi\left(\frac{\ln(t)-\bar{x}_L}{s_L}\right)$$

其中，

$$\bar{x}_L=\frac{1}{n}\sum_{i=1}^{n}\ln(t_i)\qquad s_L^2=\frac{1}{n-1}\left[\sum_{i=1}^{n}(\ln(t_i))^2-n\bar{x}_L^2\right]\tag{a}$$

如果对累积函数求反函数，则得到：

$$y=\beta_0+\beta_1x$$

式中，

$$y=\Phi^{-1}(F(t))\qquad x=\ln(t)\quad \beta_0=-\frac{\bar{x}_L}{s_L}\quad \beta_1=\frac{1}{s_L}$$

$\Phi^{-1}(\cdots)$由 norminv 函数实现，t 的平均值和方差[不是 ln(t)，ln(t)服从正态分布]

分别由下式求出：

$$\bar{x}_t = \exp(\bar{x}_L + s_L^2/2) \quad s_t^2 = (\exp(s_L^2) - 1)\exp(2\bar{x}_L + s_L^2)$$

上式可由 lognstat 函数求得。

(a) 对于表 14.24 中所列数据，利用例 14.3 所介绍的方法判断其是否服从对数分布。表中数据已排好序。另外，画出以 $\ln(t)$ 为自变量 $F(t)$ 函数的关系曲线以及拟合直线，并用 normplot 函数显示出方差值。

(b) 对用图形方法得到的 x_L 和 s_L 与由式(a)得到的 x_L 和 s_L 值比较。

[答案：从拟合曲线得：$x_L = 2.507\,2$，$s_L = 0.888\,41$；由式(a)得：$x_L = 2.507\,2$，$s_L = 0.854\,41$。]

表 14.24　练习 14.11 中的数据

1 55	15 70
3.05	16.35
3.65	17.70
5 20	17.95
7 75	19 45
10 45	19 80
10 85	20.05
10 90	32.75
12 65	35 45
15.25	49 35

14.12 一元线性回归分析的相关系数 R 可由式(14.32)求出，现对下述假设做出检验：

$$H_0: R = 0$$

$$H_1: R \neq 0$$

建立检验统计量：

$$t_0 = \frac{R\sqrt{n-2}}{\sqrt{1-R^2}}$$

将 t_0 与 $t_{\alpha/2,n-2}$ 比较，如果 $t_0 > t_{\alpha/2,n-2}$，则放弃 H_0。实际检验的是对应 t_0 的 p 值。$n \geqslant 25$ 时，相关系数 r 的置信区间由下式求出：

$$\tanh\left(\operatorname{artanh}(R) - \frac{-z_{\alpha/2}}{\sqrt{n-3}}\right) \leqslant r \leqslant \tanh\left(\operatorname{artanh}(R) + \frac{z_{\alpha/2}}{\sqrt{n-3}}\right)$$

式中 $z_{\alpha/2} = \mathrm{norminv}(1 - \alpha/2)$[参考式(14.19)]。

对于表 14.25 所列数据：

(a) 如果回归模型为 $y = \beta_0 + \beta_1/x$，确定回归系数。

(b) 画出回归线及各个数据点。

(c) 判断残差是否为正态分布。

(d) 确定(i)相关系数是否等于 0，(ii)置信度为 95% 的置信区间。

[答案：(a) $\beta_0 = 8.936\,6$，$\beta_1 = -41.607\,3$；(d) $0.976\,998 \leqslant 0.989\,96 \leqslant 0.995\,64$。]

表 14.25　练习 14.12 中的数据

x	y	x	y
10.0	4 746	11.6	5 211
12.0	5 466	14.8	6 264
6.8	3.171	7 2	3.411
5.4	1 500	15 7	6 537
20 0	6.708	17 6	6 336
19 4	7.158	14 0	5 400
19.1	6 882	10 9	4 503
6.1	1 674	18 2	6.909
16.3	6.498	20 4	6.930
12.4	5.598	8 2	3 582
5.8	1 959	7 9	3 432
12.7	5.790	4 9	0.369
9.2	4.686		

14.13　多元线性回归分析中，经常检验的两类残差为：

(a) 标准化残差，其定义为：

$$d_i = \frac{e_i}{\hat{\sigma}}$$

其中 $\hat{\sigma}^2$ 由式(14.41)定义。

(b) 学生化残差，定义如下：

$$r_i = \frac{e_i}{\hat{\sigma}\sqrt{1-h_{ii}}}$$

式中 h_{ii} 为下式的第 i 个对角线元素：

$$\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$$

$\mathbf{X}$ 由式(14.35)定义。

用例 14.6 中的模型及表 14.7 中的数据求出其标准化残差和学生化残差。$\hat{y}_i$($= y_i - e_i$，y_i 在同一图上画出这两种残差随平均输出量为表 14.7 中的输出值)的变化图。用不同符号区别两个图形。分析这些方差值有无远距离点。在此，超过 3 就属于远距离点。用 legend 函数画出图并标出两组不同的残差值。

14.14　2^k 析因实验公式在 $n=1$ 时可做如下修改：效果 *Effects*$_k$ 的计算如第 14.6.2 节所示，将其按从正的最大值到负的最大值进行降序排序，参考例 14.9 结果表中标为"Effect"的列。排序过程中必须记录所有因素及其交互效应。已排序的效果用 normplot 函数在图上画出。可以忽略的效果(不显著的)将服从正态分布，在图中分布于一条直线上，而那些显著性效果值将明显偏离该直线。表 14.26 中的数据为重复一次的 2^4 析因实验所得数据，确定其中哪些值为显著的。可以从 normplot 函数得到的图中直观地看出，也可由已排序的效果及其对应的因素或交互效应的列表中直观地看出。

[答案:*A*,*C*,*D*,*AC*,*AD* 占总平方和的 96.6%,对因素和交互效应进行由正到负的排序:A,C,B,BCD,BC,ABC,ACD,CD,BD,AB,ABCD,ABD,D,AD,AC。$Effect_A$ = 43.25,$Effect_C$ = 19.75, $Effect_D$ = −29.25, $Effect_{AD}$ = −33.25, $Effect_{AC}$ = −36.25。]

表 14.26 重复 $n=1$ 次的 2^4 析因实验数据

运行序号⁺	y_m	运行序号⁺	y_m
1	86	9	90
2	200	10	142
3	90	11	96
4	208	12	130
5	150	13	136
6	172	14	120
7	140	15	160
8	192	16	130

⁺运行序号与表 14 14 中的水平组合相一致

参考文献

1. Barker T B. Quality by Experimental Design. New York: Marcel Dekker, 1985.
2. Box G E P, W G Hunter, and Hunter J S. Statistics for Experimenters. New York: John Wiley & Sons, 1978.
3. Breyfogle Ⅲ F W. Statistical Methods for Tesing, Development and Manufacturing, New York: John Wiley & Sons, 1992.
4. Draper N and Smith H. Applied Regression Analysis. 2nd ed. New York: John Wiley & Sons, 1981.
5. Elsayed E A. Reliability Engineering. MA: Addison Wesley Longman, Inc. Reading, 1996.
6. Frigon N L and Mathews D. Practical Guide to Experimental Design. New York: John Wiley & Sons, 1997.
7. Hayter A J. Probability and Statistics for Engineers and Scientists, PWS Publishing Co., Boston, 1996.
8. Lapin L L. Modern Engineering Statistics. CA: Duxbury Press, Belmont, 1997.
9. Lewis E E. Introduction to Reliability Engineering. 2nd ed. New York: John Whiley & Sons, 1991.
10. Montgomery D C. Design and Analysis of Experiments, 3rd ed. New York: John Wiley & Sons, 1991.
11. Montgomery D C and Runger G C. Applied Statistics and Probability for Engineers. New York: John Wiley & Sons, 1994.
12. Myers R H and Montgomery D C. Response Surface Methodology: Process and Product Optimization Using Designed Experiments. New York: John Wiley & Sons, 1995.
13. Walpole R E, Myers R H and Myers S L. Probability and Statistics for Engineers and Scientists. 6th ed. NJ: Prentice Hall, Upper Saddle Rive, 1998.